"DIGITAL COMPUTER APPLICATIONS IN A WEAPON CONTROL SYSTEM"

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

by

Lieut. A.M. BOOPATHI, I.N.

to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

August 1972

CERTIFICATE

Certified that this work on "Digital Computer Applications In a Weapon Control System" has been carried under my supervision and that this has not been submitted elsewhere for a degree.

Radhakrishnan

Dr. T. Radhakrishnan
Lecturer
Computer Centre
Department of Electrical Engg.
IIT-Kanpur

ACKNOWLEDGEMENT

I am greatly indebted to Dr. T. Radhakrishnan who has been a source of inspiration for me in completing this work. The valuable discussions I had with him and his useful suggestions in various aspects made this work a complete one. I take this opportunity to express my gratitude to a number of my friends who have given me a helping hand in my programming work and for their valuable discussions. Last but not least, I thank Mr. S.V. Sirbhai for his neat typing work.

A M Boopathi
(A.M. BOOPATHI)
LIEUTENANT
INDIAN NAVY

KANPUR-16
16th August '72.

CONTENTS

		Page
	LIST OF FIGURES	vi
	ABSTRACT	vii
CHAPTER 1	INTRODUCTION	1
1.1	General	1
1.2	General Description of the System	1
1.3	Possible Sources of Error	4
1.4	Possibility of a Correction Technique	5
1.5	Outline of the Thesis	5
CHAPTER 2	ERROR IN PREDICTION	6
2.1	Total Error in Prediction	6
2.2	Simulation of the System	8
2.21	Simulation of the Aircraft	9
2.22	Simulation of the Shell Trajectory	12
2.23	Error due to Imperfections in Simulation	14
CHAPTER 3	CORRECTION FUNCTION FOR THE ERROR IN PREDICTION	16
3.1	Analysing the Error in Prediction	16
3.2	Curve Fitting	19
3.3	The Correction Technique for the Predictor	21
3.4	Further Improvement in Performance with Modified Technique	28

	3.5	A Critical Analysis	32
	3.51	Time Delay in Determining the Correction Function Coefficients	32
	3.52	Cost of Implementation of the Technique Vs Improvement in Performance	34
CHAPTER	4	A SPECIAL PURPOSE COMPUTER FOR WEAPON CONTROL	36
	4.1	Use of the ICs in Digital Systems	36
	4.2	Selection of the Instruction Set	40
	4.3	Memory Requirements	42
	4.31	Data Format	42
	4.32	Instruction Format	43
	4.4	Input/Output Units	44
	4.5	Arithmetic Units	46
	4.6	Control Circuits	47
	4.7	Selection of Clock Rate	56
	4.8	Total Time for Execution of the Program and Power Requirements	58
	4.9	Advantages of the proposed Computer	58
CHAPTER	5	CONCLUSIONS AND SCOPE FOR FURTHER WORK	62
APPENDIX	- I	LIST OF TYPICAL CASES ANALYSED	64
APPENDIX	- II	PROGRAM TO SOLVE THE FIRE CONTROL PROBLEM	67
APPENDIX	- III	CONTROL CIRCUITS TO EXECUTE MICRO/MACRO OPERATIONS	79
APPENDIX	- IV	CIRCUITS FOR MSGs	87
LIST OF REFERENCES		...	89
GENERAL REFERENCES		...	90
GLOSSARY		...	91

LIST OF FIGURES

		Page
Figure 1.1	Block Diagram of a Simplified Weapon Control System	2
Figure 2.1	Error in Prediction	7
Figure 2.2	Shell Burst	7
Figure 2.3	Movement of Aircraft in Vertical Plane	10
Figure 2.4	Plan View of movement	10
Figure 2.5	Variation of C_d with v	13
Figure 2.6	Shell trajectory	14
Figure 3.1(a) to 3.1(d)	Variation of error in prediction with future range	17
Figure 3.2	Flow chart of the program	22
Figure 3.3(a) to 3.3(d)	Variation of Improvement factor with future range	26
Figure 3.4	Modified set of data for curve fitting	29
Figure 3.5(a) to 3.5(c)	Variation of improvement factor for various N	29
Figure 3.6	Variation of Improvement factor with various number of samples	31
Figure 4.1	Flow diagram of the computer organisation	39
Figure 4.2(a)	Circuit for Instruction Fetch	48
Figure 4.2(b)	Timing Pulses for Instruction Fetch	48
Figure 4.3	Opcode decoding circuit	5[illegible]

ABSTRACT

A predictor correction technique to improve the performance of Weapon Control Systems has been developed and found to work satisfactorily for various typically chosen cases. The logical organisation of a special purpose Digital Computer to be used with the Weapon Control System is described.

This is organised in two parts. In the first part the predictor correction technique is developed and in the second part the logical organisation of a special purpose computer is described.

Chapter 1 gives the general picture of the fire control problem that is solved by the predictor in the Weapon Control System. In Chapter 2, the possible sources of error in prediction are discussed. In Chapter 3, the error in prediction is analysed and an error correction technique is deviced. In Chapter 4, the logical organisation of a special purpose computer is described. A gate level design of the computer is given and its performance compared with an existing electro-mechanical computer.

PART - I

A PREDICTOR CORRECTION TECHNIQUE

CHAPTER 1

INTRODUCTION

1.1 General

In weapon control systems dealing with mobile targets, a predictor is used to compute the angle at which a projectile is to be fired so as to destroy the moving target. This predictor is being provided with information regarding the present position and rate of change of position of the target. The predictor assumes that the target. will maintain its present course, that is, it will continue to move in the same direction at the same speed as now, and predicts its future position and the angle at which a projectile is to be fired so that the target and the projectile being fired at this moment, will collide after certain time interval. This time interval, t_f is called the "time of flight of the projectile".

The future position of the target and the time of flight of the shell are interdependant. Hence the predictor in effect solves a set of simultaneous equations. Now a simplified weapon control system using a predictor will be considered.

1.2 General Representation of the System

A simplified weapon control system used to control an anti-aircraft gun is shown in Fig.1.1. To simplify the system further, the following assumptions are made in the beginning:

1) The gun and Radar Antenna are mounted on a stabilised platform.

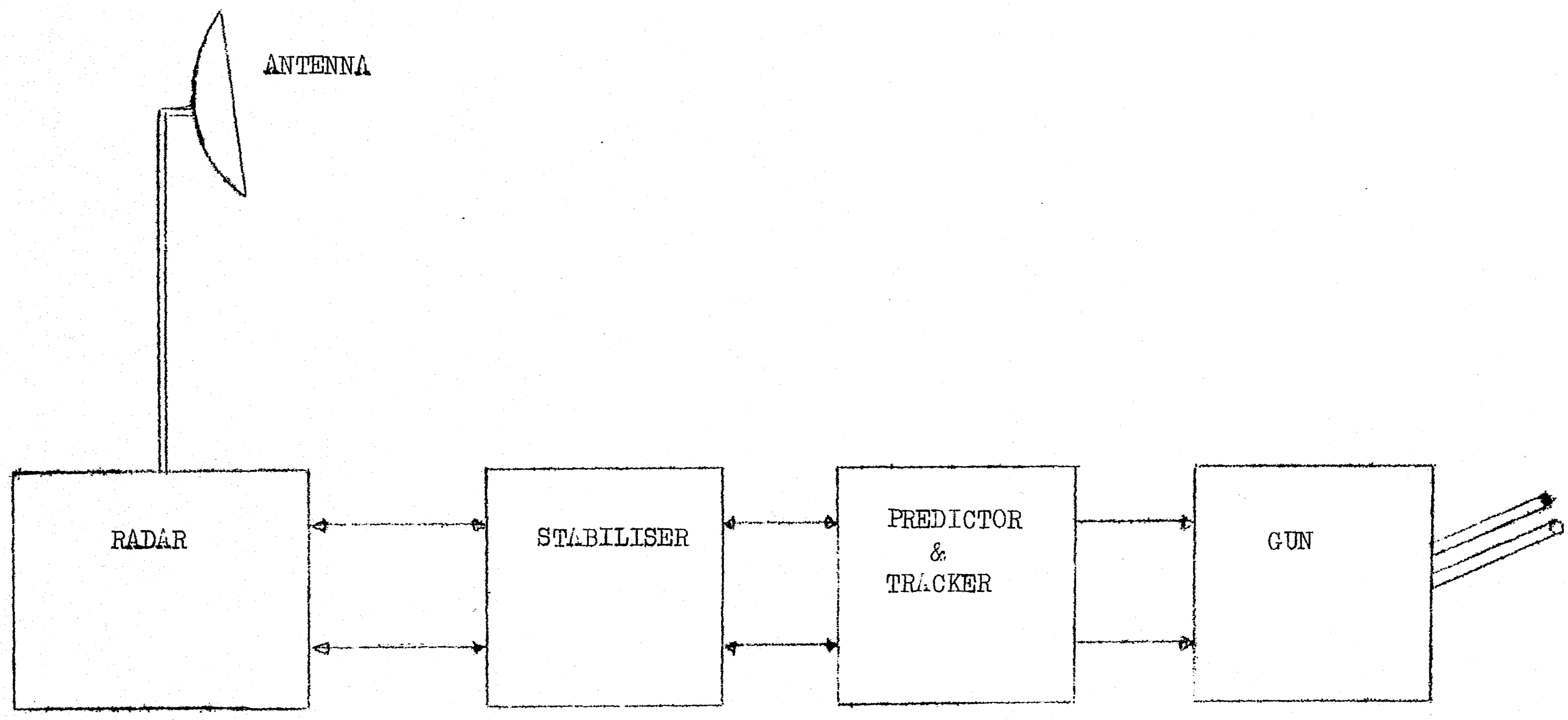


Fig.1.1 - Block Diagram of a Simplified Weapon Control System.

2) The Gun and the Radar Antenna are theoretically at the same point and plane.

These assumptions will be taken care off in the following manner. The first assumption is taken care off by applying the correcting signals from a Gyro Stabiliser and the second assumption by applying lateral and vertical convergence corrections to the angle at which the projectile is to be fired. It is also assumed that the aircraft will continue to fly in the same course. This will be justified in a later section.

When an aircraft is detected, the Radar will automatically track the aircraft. Necessary signals for this automatic tracking are generated by the Tracker. While tracking the aircraft automatically, the following informations are obtained:

1) Present range of the aircraft, R_p.
2) Present bearing of the aircraft, T_s.
3) Present elevation of the aircraft, S_p.
4) Rate of change of range of the aircraft, $\dot{R}$.
5) Rate of change of bearing of the aircraft, $\dot{B}_1$.
6) Rate of change of elevation of the aircraft, $\dot{S}_p$

These signals are fed to the predictor. The wind velocity and direction of wind are also fed to the predictor to apply necessary wind corrections. The muzzle velocity, that is, the velocity of the projectile at the time of leaving the Gun, and the ballistics of the Shell which form

another set of inputs to the predictor are manually set. The muzzle velocity and ballistics vary for different types of Guns and Shells. From these inputs the predictor is required to compute the following:

1) Angle in the horizontal plane to which the Gun should be trained before firing a projectile, T_g
2) Angle of elevation of the Gun, E_g
3) Time of flight of the Shell, t_f to set the fuze for the Shell.

The function of the predictor is to compute E_g, T_g and t_f from the available inputs. This problem is called as the fire control problem. This is represented by a set of simultaneous equations. These equations are not given here as they resemble an existing weapon control system.

1.3 Possible Sources of Error

In this weapon control system, to simplify the function of the predictor, many approximations are made. In the set of simultaneous equations used to solve the fire control problem these are incorporated. These equations involve certain constants whose values depend upon the bearing, altitude, speed and range of the aircraft and the wind conditions at that time.

Though in a real situation they are not constants but for simplifying the predictor, they are assumed to be constants. They are set at their optimum values which are determined after trials. Such imperfections

introduce error in the system and reduce the accuracy of the system. Due to this, the probability of 'kill' is reduced considerably in practice.

1.4 Possibility of a Correction Technique

When a digital computer is used alongwith the weapon control system, the predicted future positions of the aircraft at regular time intervals can be stored to find the error in prediction at various time intervals. If the variation of the error is uniform, these error values can be used to apply a correction to the predictor as explained below. If the error in prediction of the range is DR_{fi} when the predicted future range is R_{fi} for various i, then DR_{fi} can be expressed as a function of R_{fi}. When the coefficients of this function are known, for any predicted future range R_{fx}, the error in prediction DR_{fx} can be computed. This is applied as a correction to R_{fx} to reduce the error in prediction of R_{fx}. This method of applying corrections is explained in the next two chapters.

1.5 Outline of the thesis

The following is done as part of this thesis work:

1) Simulation of the Weapon Control System on a Digital Computer.
2) Development of an error correction technique to improve the performance of the predictor and hence the overall system.
3) System design of a special purpose Digital Computer to be used with this Weapon Control System.

The development of the correction technique is arranged as part I and the system design of the Special Purpose Computer as part II.

CHAPTER 2

ERROR IN PREDICTION

As mentioned earlier, error in prediction might depend on many factors like the bearing, speed and elevation of the aircraft and the wind conditions. This error is made up of two parts, namely,

1) Error in prediction of future position of the aircraft.
2) Error in prediction of the Shell trajectory and Shell position or the time of flight of the Shell.

These are shown in Fig.2.1 and will be explained below.

2.1 Total Error in Prediction

The Radar and the Gun are at the same point G, as assumed earlier. At the time, the aircraft is detected, the CLOCK 's set to zero and the aircraft position is indicated by I. Depending on the present rates of the aircraft, the predictor predicts that if a Shell is fired with an initial velocity P_v at an elevation of E_g, the Shell and the aircraft will meet at P after a time lapse of t_f (1) where $t_f(1)$ is the predicted time of flight of the Shell. The predicted future position, P and the time of flight of the Shell $t_f(1)$ are stored. As the sampling rate is chosen to be 100 msec as explained in a later section, the clock is incremented by 100 msec and the future position and time of flight of the Shell $t_f(2)$ are computed and stored. This is repeated for every 100 msec.

When the clock equals $t_f(1)$, the radar inputs are sampled. Let the aircraft be in position A. Hence PA gives the error in the prediction

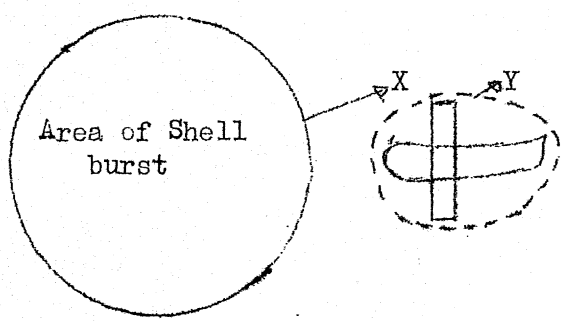


Fig. 2.2 Shell Burst

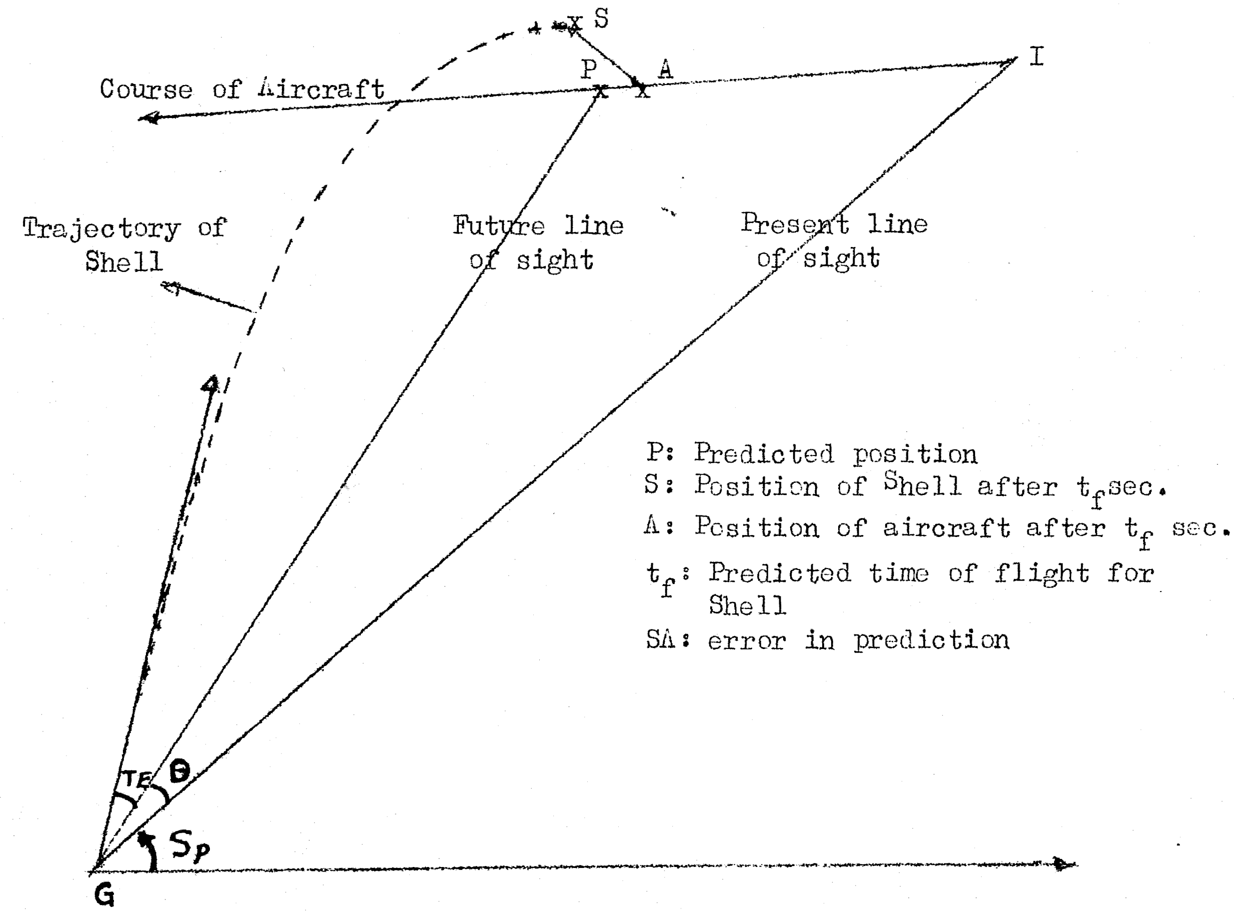


Fig.2.1: Error in Prediction

of the future position of the aircraft. At this time, the position of the Shell should be determined. But it is difficult to keep tracking a Shell with a Radar. Also normally a Shell will not be fired untill the correction function parameters are available. Hence the Shell trajectory and its position will be determined mathematically, from the aerodynamics and wind conditions. Let the position of the Shell, as determined by this mathematical model be S. Thus PS gives the error in the prediction of the Shell position. The total error in prediction is SA.

When the Shell bursts at point S, the splinters from the Shell will be sprayed over an area X as shown in the Fig. 2.2. This area is called the destructive area of the Shell. If the body of the aircraft is anywhere inside this area, that is, if the areas X and Y in the figure intersect, the aircraft will be hit. To achieve this objective, we will have to reduce the error in prediction to within $\pm$ 3 yards. This means an accuracy of the order of 0.01% at a range of 30000 yards.

We have assumed a sampling rate of 0.1 sec which needs justification Even an aircraft flying at a speed of 1 Mach will cover only about 35 yards in 0.1 sec time interval. Hence when an aircraft is in the range of 5000 yards to 30000 yards, samplings in steps of about 50 yards will be quite sufficient. Hence the sampling interval is chosen to be 0.1 sec.

2.2 Simulation of the System

This weapon control system is simulated on a Digital Computer and its performance studied. The predictor is represented by a set of

simultaneous equations. As mentioned earlier, a stable platform is assumed for the time being, hence the stabilizer is eliminated. Now the problem is to simulate the aircraft (or the Radar) and the Shell movement. These are explained below.

2.21 Simulation of Aircraft

The initial position at which the aircraft is detected, is chosen at random with random number generators, to suit the typical cases that might occur in a real system. The speed and course, i.e. direction in which the aircraft moves, are also chosen. It is assumed that the aircraft maintains its initial speed and course. This assumption will be justified later. Whenever there is a change in either the speed of the aircraft or its course, the system is initialised.

If v is the speed of the aircraft, it moves a distance of v x dt during dt, the sampling time (interval of 0.1 sec). Referring to figures 2.3 and 2.4, F gives the position of the aircraft after the time interval of 0.1 sec. R_p, S_p and T_s give the position of the aircraft at time t, and R_f, S_f and T_f give the position of the aircraft at time (t+dt). R_f, S_f and T_f are computed as shown below. Here,

X:- Angle between the present bearing T_s and the course of the aircraft. This angle is called the angle of approach of the aircraft.

and Y:- Angle between the horizontal and the course of the aircraft. This is called the diving angle of the aircraft.

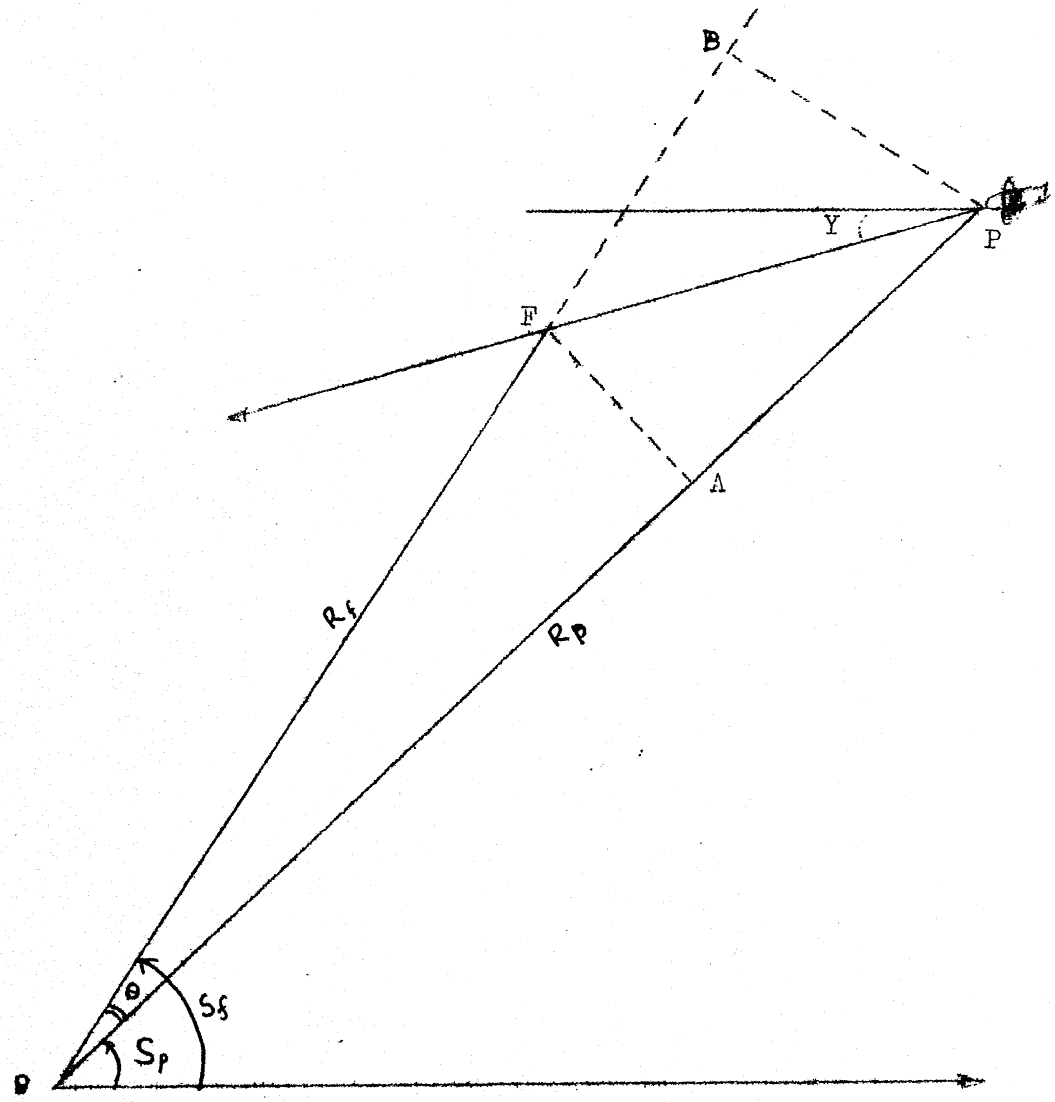


Fig. 2.3 Aircraft movement in vertical plane.

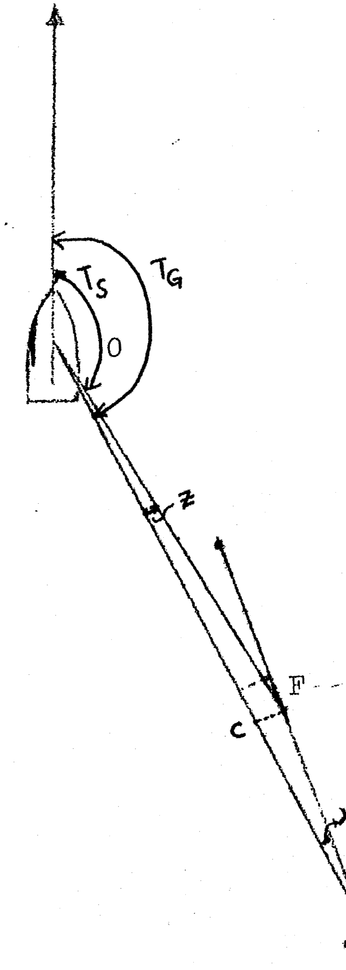


Fig.2.4: Plan View of movement

The distance moved by the aircraft in the time interval dt is greatly exaggerated in figures 2.3 and 2.4. R_f, S_f and T_s are computed as shown below.

$$R_f = R'_f \ \mathrm{Sec}\ Z' \qquad [2.1]$$

$$S_f = S_p + \theta \qquad [2.2]$$

$$T_f = T_s + Z \qquad [2.3]$$

Now as θ is very small,

$$\theta = \tan\theta = \frac{AF}{OA}$$

$$= \frac{AF}{R_p - AP}$$

$$= \frac{v.dt.\ \mathrm{Cos}X.\mathrm{Sin}\ (S_p - Y)}{R_p - v.dt.\ \mathrm{Cos}X.\ \mathrm{Cos}\ (S_p - Y)}$$

$$R'_f = OF = OB - FB$$

$$= R_p.\ \mathrm{Cos}\ \theta - v.\ dt.\ \mathrm{Cos}X.\ \mathrm{Cos}\ (S_f - Y)$$

Also as Z' is very small,

$$Z' = \tan Z'$$

$$= \frac{CF}{OC}$$

$$\doteq \frac{v.\ dt.\ \mathrm{Sin}X.\ \mathrm{Cos}Y}{R'_f}$$

Deflection on horizontal plane, Z is given by,

$$Z = \frac{Z'}{\mathrm{Cos}\ S_p}$$

$$= \frac{v.\ dt.\ \mathrm{Sin}X.\ \mathrm{Cos}Y}{R'_f\ .\ \mathrm{Cos}\ S_p}$$

2.22 Simulation of the Shell Trajectory

The Shell moves in super sonic speed and is influenced by the following forces:

1) Acceleration due to gravity, g
2) Parasite drag due to air resistance
3) Drag force due to shock waves and skin friction
4) Wind effects
5) Gyroscopic effects on the shell.

Of these, the second and third are combined to get the total drag coefficient. The total drag force acts in a direction opposite to the direction of motion of the Shell. This can be represented as,

$$F_d \propto C_d . A . V^2 \qquad [2.4]$$

where C_d :- Combined drag coefficient

V :- Velocity of the Shell

and A :- Area of cross-section of the Shell.

The combined drag coefficient C_d is a function of the Shell velocity[2].

$$\text{i.e.,} \quad C_d = f(V) \qquad [2.5]$$

C_d can be experimentally determined for various Shell velocities in a wind tunnel for any particular Shell. The variation of C_d with V is given in Fig. 2.5.

During the flight of the Shell, the velocity of the Shell varies continuously. Hence C_d would also vary with time and it will not remain constant. The velocity of the Shell at any instant varies with the acceleration on the Shell. The acceleration (or retardation) depends on

the total drag force which in turn depends on C_d. But C_d is a function of V. Thus a looping occurs as C_d and V are inter-dependant.

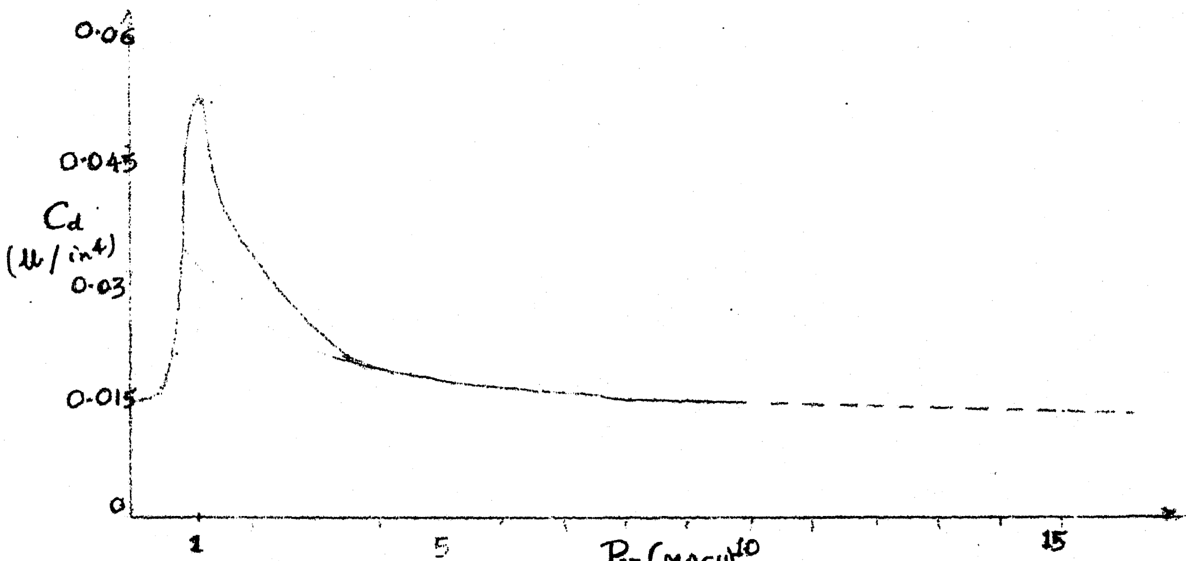


Fig. 2.5 Variation of C_d with Pv

To solve this problem we will break the total time of flight of the Shell into a number of small equal intervals. as shown in Fig. 2.6. During each small time interval, C_d is assumed to remain constant. Having fixed C_d, the total force on the Shell and hence the velocity of the Shell at the end of this time interval can be computed. For each time interval C_d will be different and is computed from the velocity of the Shell in the beginning of that time interval.

Now the number of such small intervals are to be chosen. To get this, for the same Gun elevation, time of flight and muzzle velocity, number of equal time intervals such as ~~of~~ 4,6,8,10,12 and 16 were assumed and the position

of the Shell at the end of the time of flight was computed. It has been found that there is little variation in the computed Shell position when the number of time intervals are more than 8. Hence the number of equal time intervals are chosen to be 8.

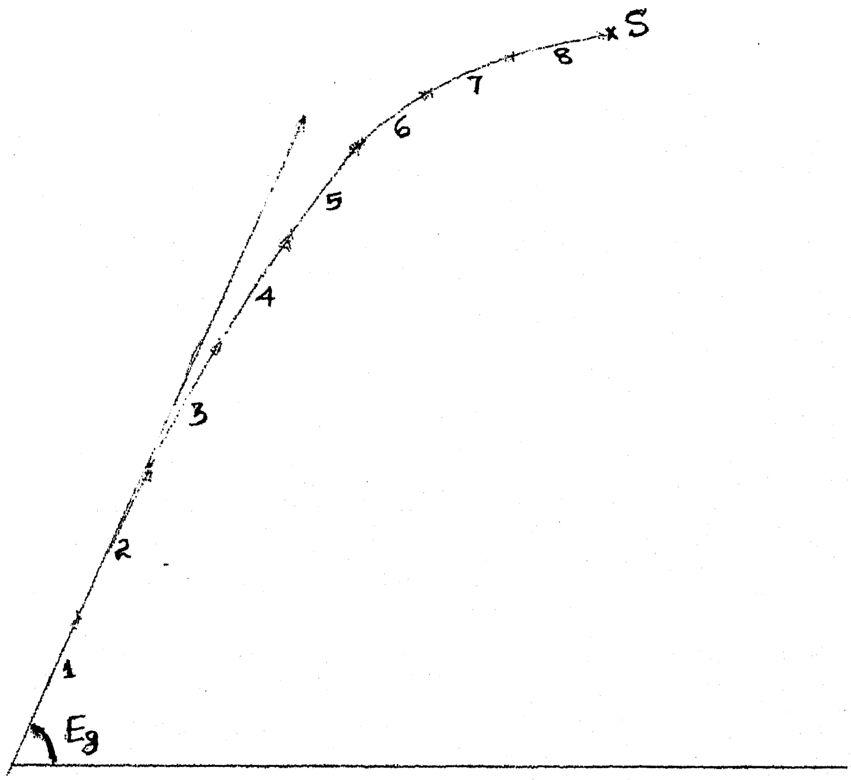


Fig. 2.6. SHELL TRAJECTORY.

2.23 Error due to Imperfections in Simulation

The mathematical model used to simulate the aircraft is an approximation to the real system and this introduces error in the solution. Similarly the analytical determination of the trajectory of the Shell also would deviate from the actual situation. Hence the error computed as shown in section 2.1 between the Shell position S and the aircraft

position A (Refer to Fig.2.1) obtained from the simulation program would comprise of the following:

1) Error in the determination of the trajectory of the Shell.

2) Error due to the non-idealness of the aircraft simulator.

3) Error in prediction.

It is felt that the first error can be reduced to a negligible value by collecting sufficient data regarding the Shell trajectories through a number of experiments and trials under varrying circumstances. In a real weapon control system, aircraft simulator will be absent and the aircraft positions are obtained from the Radar. Thus the error 2 will be absent in a real system. This leaves the error in prediction as the major component of the error in the real system.

Earlier it was stated that the predictor assumes that the aircraft maintains its speed and course. Thus the predicted position becomes invalid if the aircraft changes either the speed or its course. There are some predictors which assume a curved path that can be represented by a mathematical expression. But such predictors will fail if the aircraft takes a straight line course. Thus we have to compromise between these two different approaches to the solution of the fire control problem. In a real situation, the aircraft will pose more danger when it maintains a steady straight line course. Hence the predictor for the weapon control system under consideration is designed to tackle the aircraft effectively when the aircraft takes a steady straight line course. Thus the earlier assumption is justified. However a better system using a gradient approach method for the predictor can be designed.

CHAPTER 3

CORRECTION FUNCTION FOR THE ERROR IN PREDICTION

3.1 Analysing the Error in Prediction

In the previous chapter, we have seen that the error in the system is mainly due to the error in prediction. To improve the performance of the system, the error in prediction is to be minimised. To develope some means of reducing this error, the variation of this error under various conditions are studied. The error might vary with the following main parameters:

1) Present position of aircraft

2) Speed of the aircraft

3) Wind effects

To take into account, all the possibilities that might occur in real situations, about fifty randomly selected typical cases have been analysed. These cases are tabulated in Appendix - I. For each case the following variations of error in prediction with predicted future range is studied.

1) DR Vs R_f (R: Range)

2) DB Vs R_f (B : Bearing)

3) DE Vs R_f (E : Elevation)

Some of these variations are shown in figures 3.1(a) to 3.1(d). From these figures, the following points have been observed:

1) The variation of DB with R_f is mostly linear.

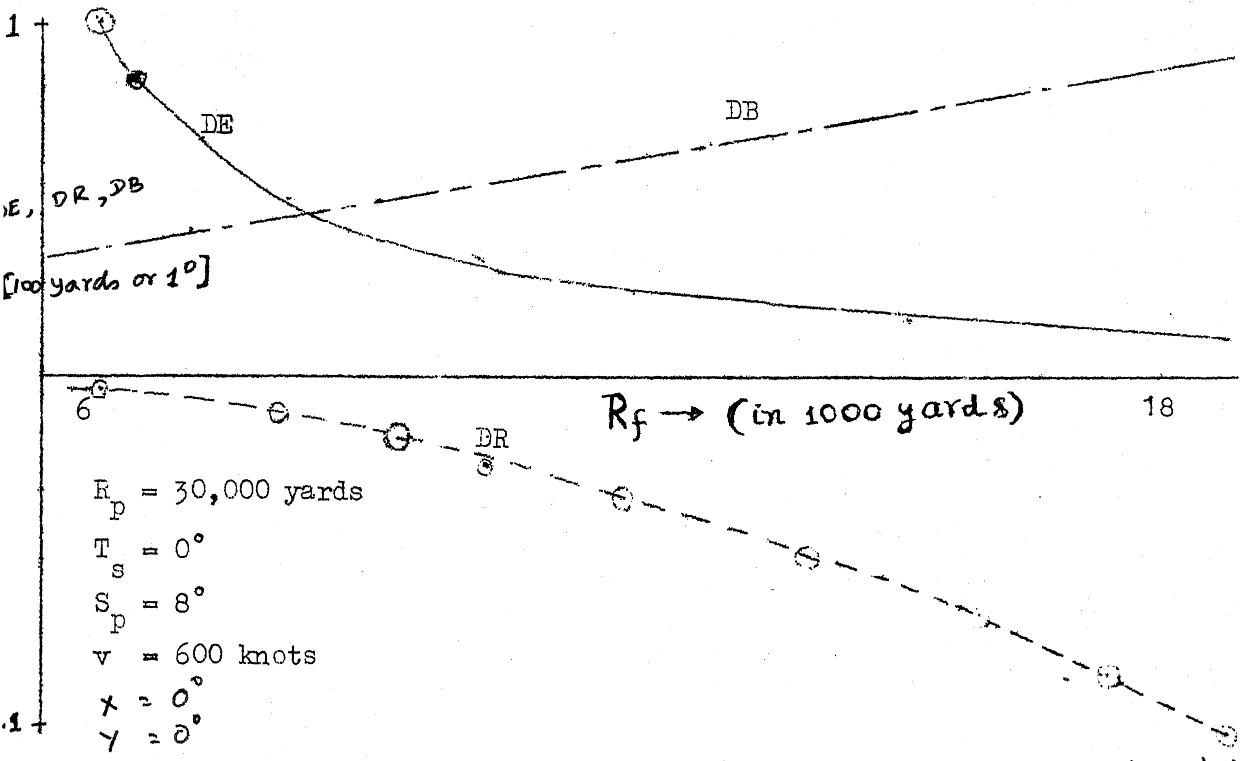


Fig. 3.1(a) Variation of error in prediction with range.

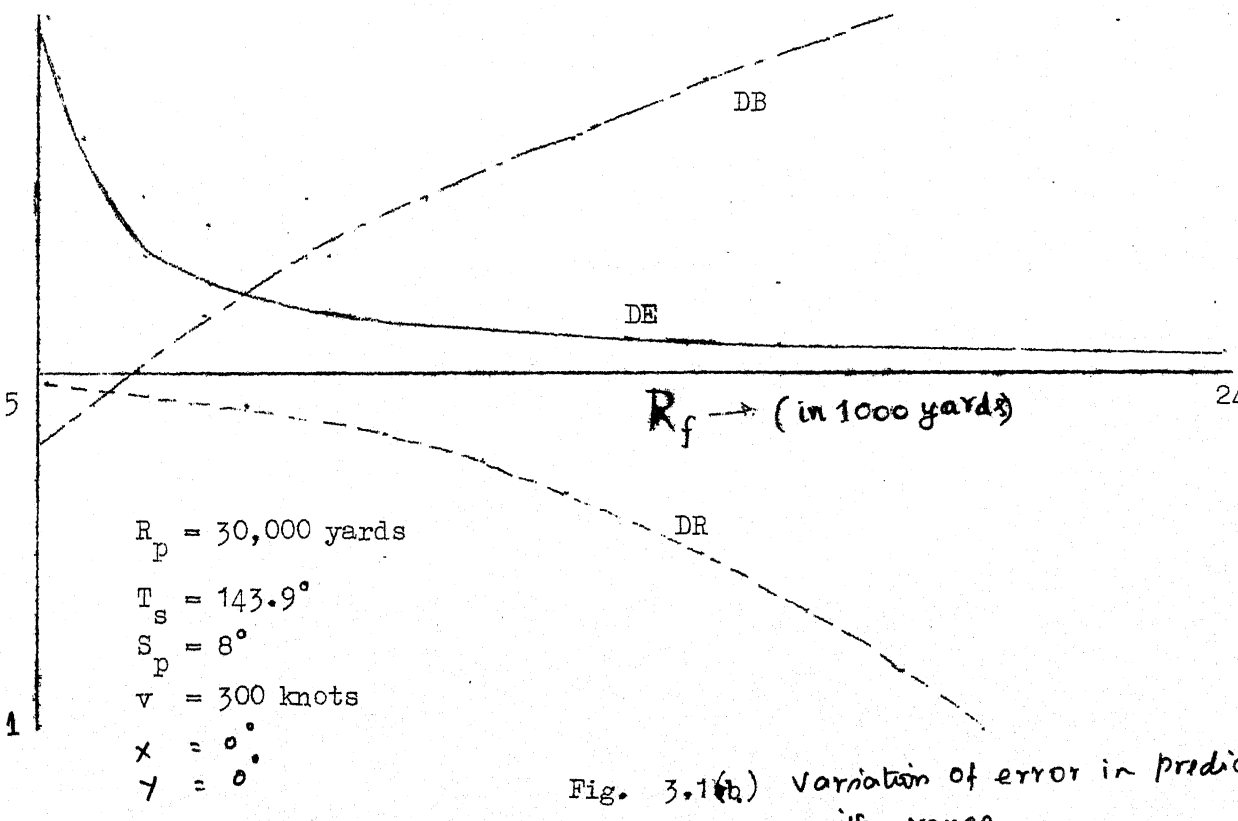
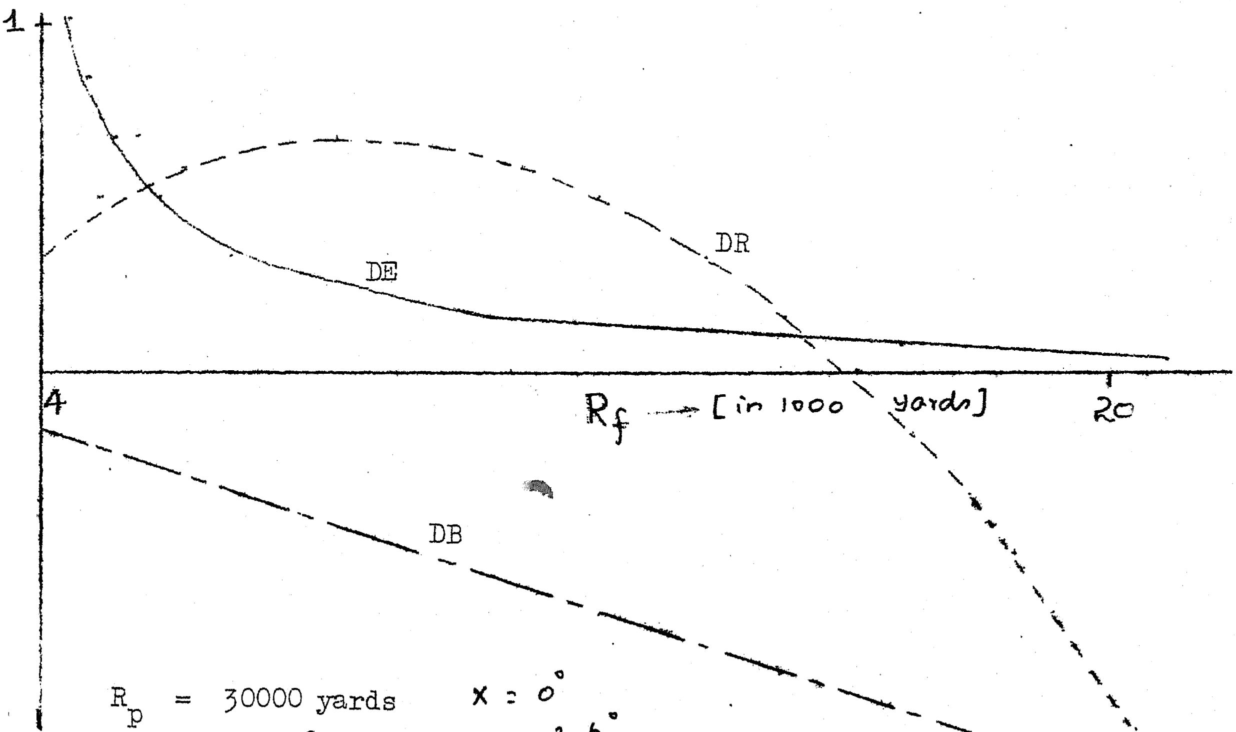
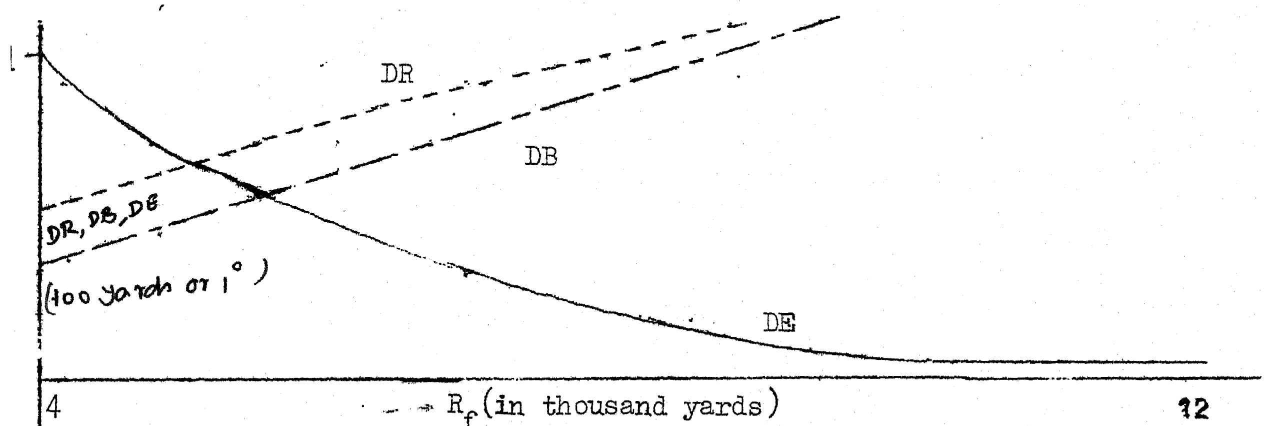


Fig. 3.1(b) Variation of error in prediction with range.



R_p = 30000 yards X = 0°
T_s = 340.9° Y = -3.6°
S_p = 8° Aircraft diving at an angle of 3.6°
v = 300 knots

Fig. 3.1(c)



R_p = 17123.5 yards
T_s = 127.1°
S_p = 2°
v = 300 knots
X = 0°
Y = 0°

Fig. 3.1(d)

Fig. 3.1(c) and Fig 3.1(d) Variation of error in prediction with range.

2) The variation of DR with R_f is nonlinear.

3) The variation of DE with R_f is nonlinear.

4) The variation of all the above mentioned errors with R_f is smooth.

These observations indicate the possibility of fitting curves to represent the variation of errors with R_f and extrapolate them to find the errors in future predictions.

3.2 Curve Fitting

The variation of DB with R_f is fairly linear as seen from the Fig.3.1(a) to 3.1(d). Thus a straight line of the form

$$DB = A_o +, A_1 . R_f \qquad [3.1]$$

can be fitted[1] to represent the variation of DB with R_f. For ten predicted future ranges R_{f1} through R_{f10}, error in bearing DB_1 through DB_{10} are obtained. With these ten sets of data, the parameter A_o and A_1 are determined using linear least squares method. of curve fitting. Now, for any future predicted range R_{fx}, probable error in bearing will be DB_x as given by,

$$DB_x = A_o + A_1 . R_{fx} \qquad [3.2]$$

Hence, if B_x is the predicted bearing, this is corrected by an amount DB_x. Thus the angle to which the Guns are to be trained before firing is given by the following equation:

$$\begin{aligned} B_{xc} &= B_x - DB_x \\ &= B_x - A_o - A_1 . R_{fx} \qquad [3.3] \end{aligned}$$

The variation of DE with R_f and DR with R_f are non-linear. They

can be conveniently represented by polynomials of the form,

$$DE = C_o + C_1 . R_f + C_2 . R_f^2 + \quad . . . \qquad [3.4]$$

$$DR = D_o + D_1 . R_f + D_2 . R_f^2 + \quad . . . \qquad [3.5]$$

To simplify the solution, only the first three terms of the polynomials are considered. Then the above equations will become,

$$DE = C_o + C_1 . R_f + C_2 . R_f^2 \qquad [3.6]$$

$$DR = D_o + D_1 . R_f + D_2 . R_f^2 \qquad [3.7]$$

These equations are linearizable as shown below[1]. Rewriting the equations,

$$DE = C_o + C_1 . R_f + C_2 . R_s \qquad [3.8]$$

$$DR = D_o + D_1 . R_f + D_2 . R_s \qquad [3.9]$$

where $R_s = R_f^2$

These equations for DE and DR are linear in R_f and R_s and their coefficients are computed as shown below:

$$A_o = \overline{DB} - A_1 . \overline{R}_f \qquad [3.10]$$

$$A_1 = \Sigma (R_{fi} - \overline{R}_f) . DB_i \qquad [3.11]$$

$$C_2 = \frac{(EE . DR - EA . AE)}{DF} \qquad [3.12]$$

$$C_1 = \frac{(ER . EA - EE . AE)}{DF} \qquad [3.13]$$

$$C_o = \overline{DE} - C_1 . \overline{R}_f - C_2 . \overline{R}_s \qquad [3.14]$$

$$D_2 = \frac{(RE . DR - RA . AE)}{DF} \qquad [3.15]$$

$$D_1 = \frac{(ER.\ RA - RE\ .\ AE)}{DF} \qquad [3.16]$$

$$D_o = \overline{DR} - D_1\ .\ \bar{R}_f - D_2\ \bar{R}_s \qquad [3.17]$$

and $$DF = (\ ER.\ DR - AE\ .\ AE) \qquad [3.18]$$

where $$DR = \sum (R_{fi} - \bar{R}_f).\ R_{fi} \qquad [3.19]$$

$$ER = \sum (R_{si} - \bar{R}_s)\ .\ R_{si} \qquad [3.20]$$

$$EA = \sum (R_{fi} - \bar{R}_f\ )\ .\ DE_i \qquad [3.21]$$

$$EE = \sum (R_{si} - \bar{R}_s\ ).\ DE_i \qquad [3.22]$$

$$RE = \sum (R_{si} - \bar{R}_s\ ).\ \ DR_i \qquad [3.23]$$

$$RA = \sum (R_{fi} - \bar{R}_f\ )\ .\ DR_i \qquad [3.24]$$

$$AE = \sum (R_{fi} - \bar{R}_f\ ).\ R_{si} \qquad [3.25]$$

$\overline{DR}$: Mean of DR

$\overline{DE}$: Mean of DE

$\bar{R}_f$: Mean of R_f

and $\bar{R}_s$: Mean of R_s

3.3 The Correction Technique for the Predictor

A flow chart depicting the correction technique is presented in fig.3.2. Referring to the flow chart, when an aircraft is detected, the system is initialised and the real time CLOCK is set to zero. The Radar inputs are sampled and the predictor computes the future position of the aircraft and the time of flight of the Shell. The time of flight t_{f1} is added to the CLOCK and stored as $t_f(1)$. The trajectory of the Shell is

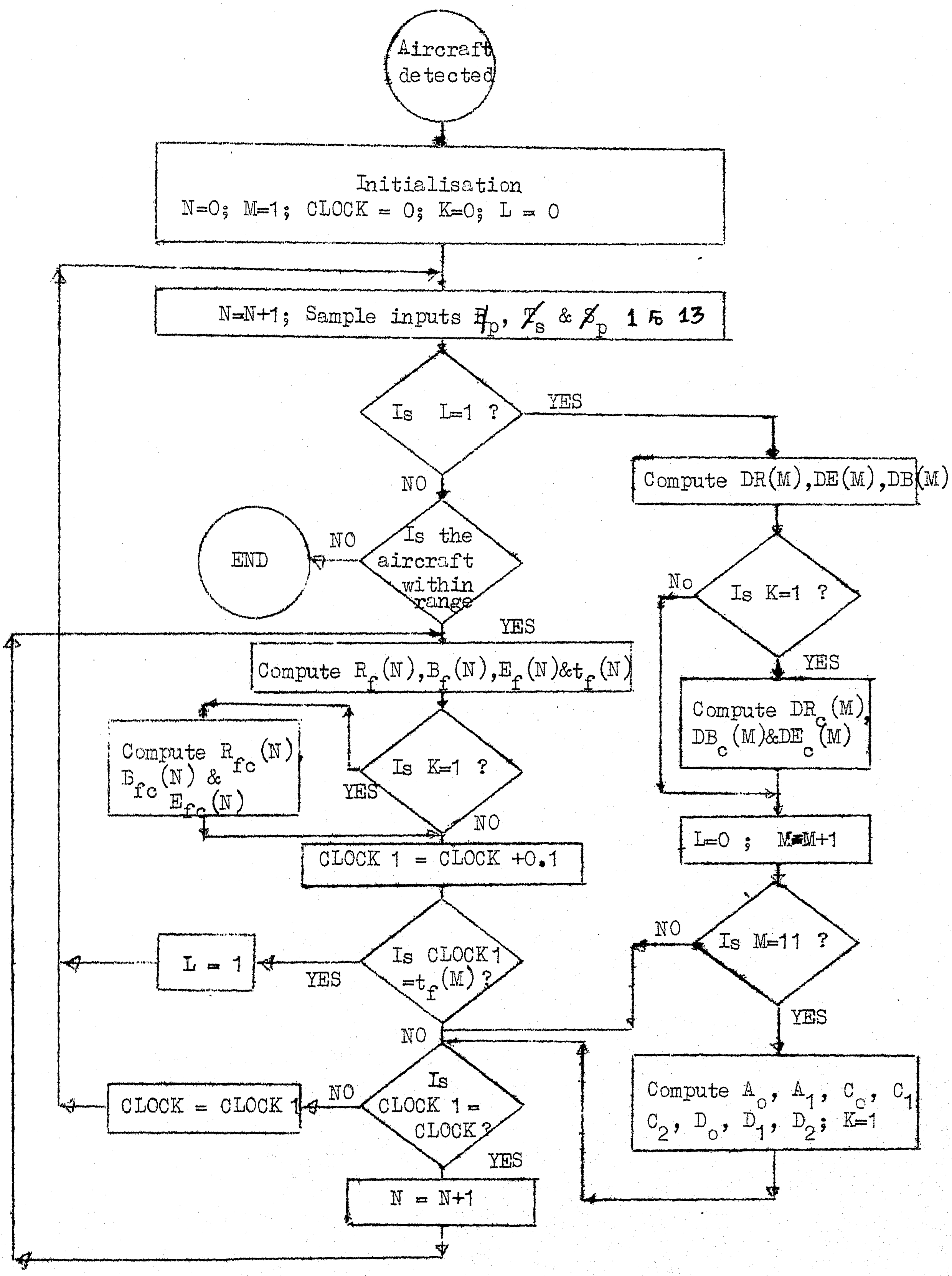


Fig. 3.2 - FLOW CHART OF THE PROGRAM

determined and its position after t_{f1} seconds, is computed as explained in section 2.2. This position in terms of range, bearing and elevation are stored as $R_f(1)$, $B_f(1)$ and $E_f(1)$ respectively. Now, after 100 msec. i.e when the CLOCK equals 100 msec., the inputs are sampled again and the future position and time of flight of the Shell, t_{f2} computed and stored as $R_f(2)$, $B_f(2)$, and $E_f(2)$. The CLOCK is added to t_{f2} and stored as $t_f(2)$. This is repeated for every 100 msec time interval as we have chosen the sampling interval to be 100 msec.

When the real time CLOCK equals $t_f(1)$, the radar inputs are sampled to find the actual position of the aircraft as given by R_p, T_s and S_p. These are compared with $R_f(1)$, $B_f(1)$ and $E_f(1)$ respectively to get the errors DR (1), DB (1) and DE (1). Again, the inputs are sampled when the CLOCK equals $t_f(2)$ to get the errors DR(2), DB(2) and DE(2) and this process is repeated.

A set of **ten** values of DR, DE and DB are collected for fitting curves between the following pairs of variables:

1) DR Vs R_f
2) DB Vs R_f
3) DE Vs R_f

These curves will be used as correction functions. Now onwards, for each input sampling the future positions $R_f(x)$, $B_f(x)$ and $E_f(x)$ as well as their corrected values $R_{fc}(x)$, $B_{fc}(x)$ and $E_{fc}(x)$ are computed and stored. If the aircraft is within the firing range of the Gun, the firing can

commence with the corrected values of the future position i.e. $R_{fc}(x)$, $B_{fc}(x)$ and $E_{fc}(x)$.

As before, whenever CLOCK equals $t_f(x)$ the inputs are sampled to get the actual position of the aircraft at that time. These are given by R_p, T_s and S_p. Now the errors in prediction of range, bearing and elevation with and without correction being applied are computed as indicated below.

$$DR(x) = R_f(x) - R_p \qquad [3.26]$$

$$DE(x) = E_f(x) - S_p \qquad [3.27]$$

$$DB(x) = B_f(x) - T_s \qquad [3.28]$$

$$DR_c(x) = R_{fc}(x) - R_p \qquad [3.29]$$

$$DE_c(x) = E_{fc}(x) - S_p \qquad [3.30]$$

$$DB_c(x) = B_{fc}(x) - T_s \qquad [3.31]$$

where, $DR(x)$, $DE(x)$ and $DB(x)$ are error in prediction of range elevation and bearing respectively without the correction being applied.

$DR_c(x)$, $DE_c(x)$ and $DB_c(x)$ are the error in range, elevation and bearing respectively with the application of correction technique.

$R_{fc}(x)$, $E_{fc}(x)$ and $B_{fc}(x)$ give the corrected future position.

The variation of the error in prediction with and without the application of the correction technique has been studied for forty eight typically chosen cases. These cases are tabulated in Appendix I. The variation of the error in a few cases are already shown in Fig.3.1(a) to 3.1(d).

The improvement in the performance of the predictor due to the application of the correction technique is evaluated as indicated below:

$$F_r(x) = \frac{DR(x)}{DR_c(x)} \quad [3.32]$$

$$F_b(x) = \frac{DB(x)}{DB_c(x)} \quad [3.33]$$

$$F_e(x) = \frac{DE(x)}{DE_c(x)} \quad [3.34]$$

where $F_r(x)$, $F_b(x)$ & $F_e(x)$ are improvement factors in range, bearing and elevation respectively.

The variation of $F_r(x)$, $F_b(x)$ and $F_e(x)$ with $R_f(x)$ for four typical cases are shown in Fig. 3.3(a) to 3.3(d). After studying these variations for the 48 cases given in Appendix I, the following conclusions are drawn:

1) The error in the prediction of range is minimised (to a very small value) after the application of the correction technique. It is always less than $\pm$ 2 yards for slower aircrafts. For faster aircrafts DR_c is within $\pm$ 5 yards in many cases and in a few cases DR_c is as high as $\pm$ 15 yards.

2) The improvement in the prediction of bearing ~~and~~ and elevation are much less compared to the improvement in the prediction of range (Refer Fig. 3.3(a) to 3.3(d)).

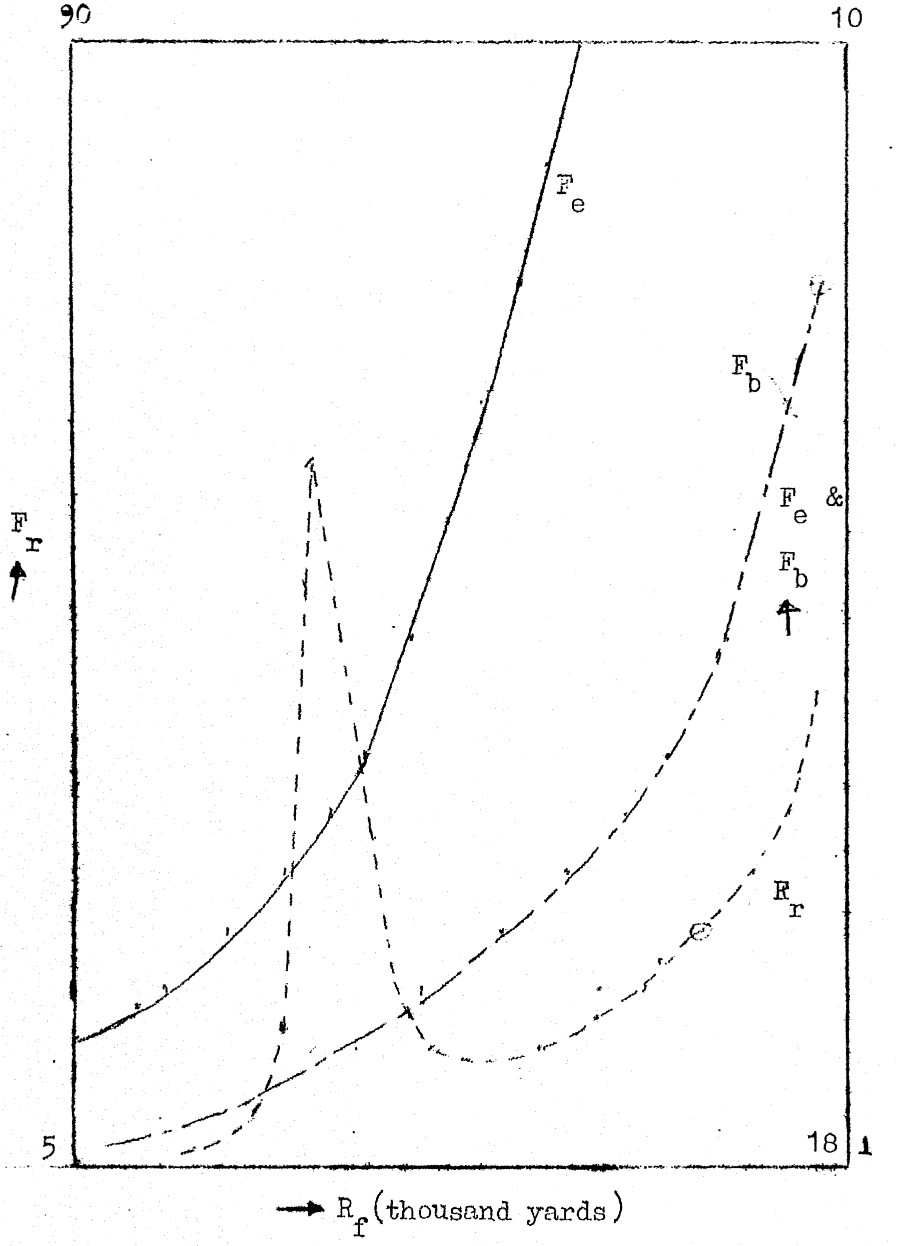


Fig. 3.3(a)

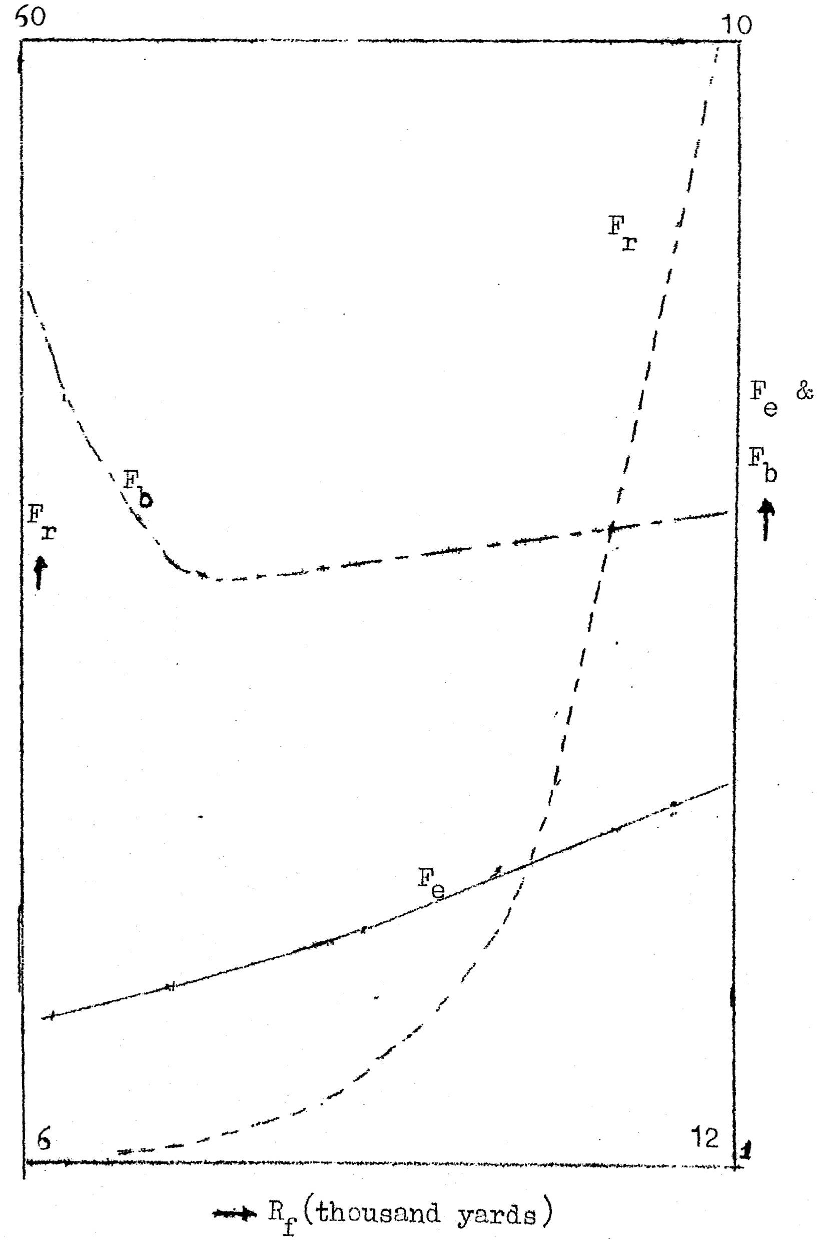


Fig.3.3(b)

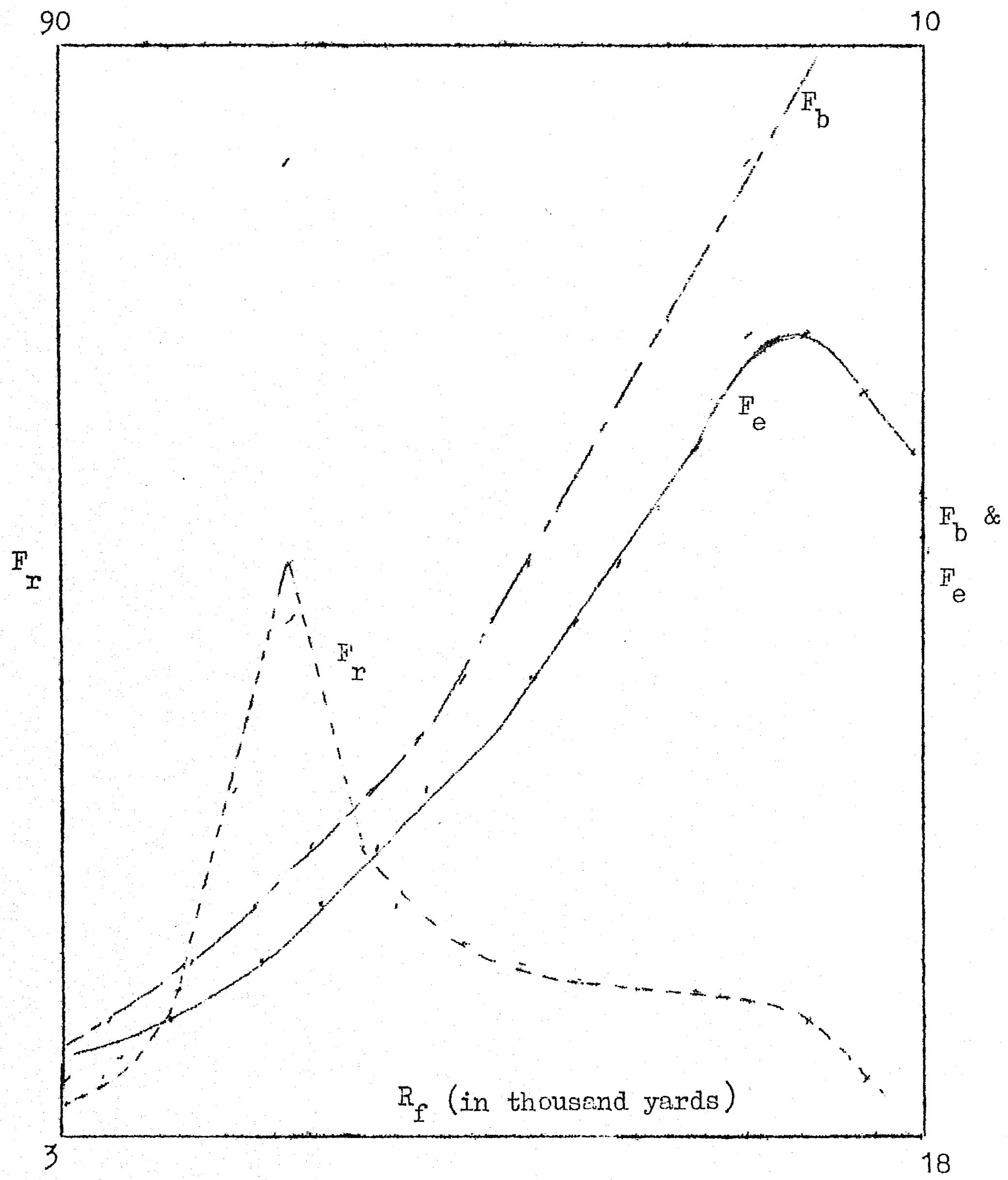


Fig. 3.3(c)

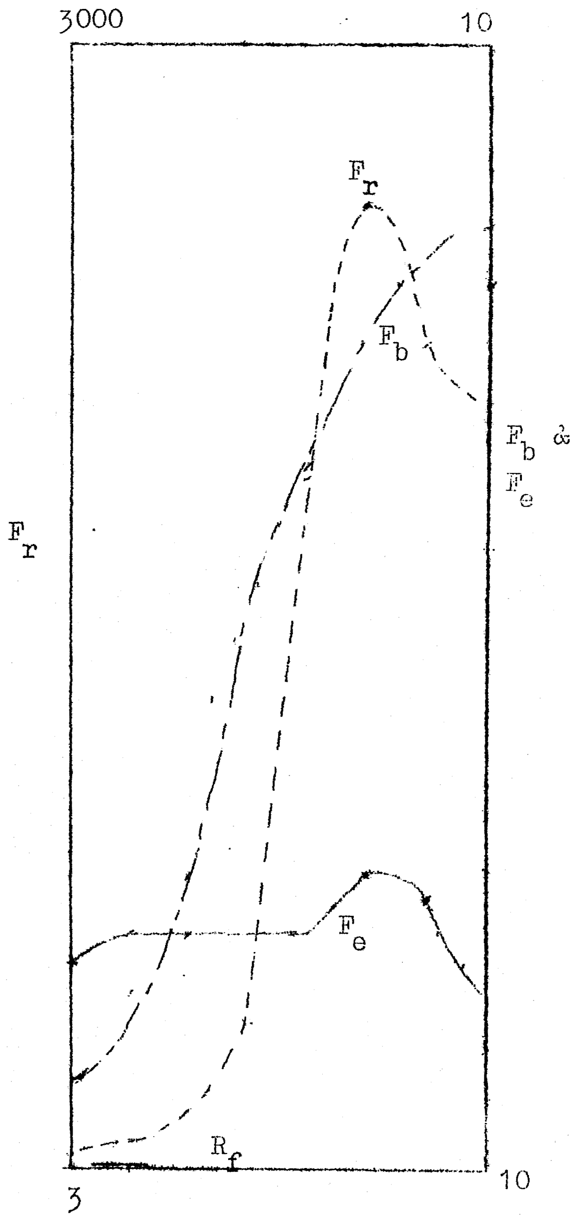


Fig.3.3(d)

(3) Soon after the correction coefficients are evaluated, the improvement in prediction is high but as time passes on the improvement factor decreases appreciably.

(4) The spikes in the curves appear due to the variables DR_c or DB_c or DE_c cross the zero axis and has no physical significance.

3.4 Further Improvement in Performance with Modified Technique

The observations made under point (3) mentioned ~~(3)~~ above indicate that the extrapolation of the correction function curve, with increasing time introduces larger disparity. To alleviate this and improve the performance of the system a modified technique of curve fitting with latest set of data points has been tried.

This method of curve fitting is represented schematically in Fig.3.4. This resembles a push down stack. Whenever N number of new samples of error values are obtained, the curve fitting routine is entered and a smooth curve is fitted with the newest 10 samples. Thus the correction function coefficients are modified after every N number of samples are obtained. This method has been tried with different N like 1, 10, 20, 40 and 80. It has been observed that there is very little improvement in the performance of the predictor when the N is small compared to larger values of N. The results are shown in Fig. 3.5.(a)to(c). A sample case with N as 100 has been chosen for investigation. It has been observed that the error in prediction with correction being applied, remains within the tolerable limits when the value of N is 100. Hence it is decided to have N as 100.

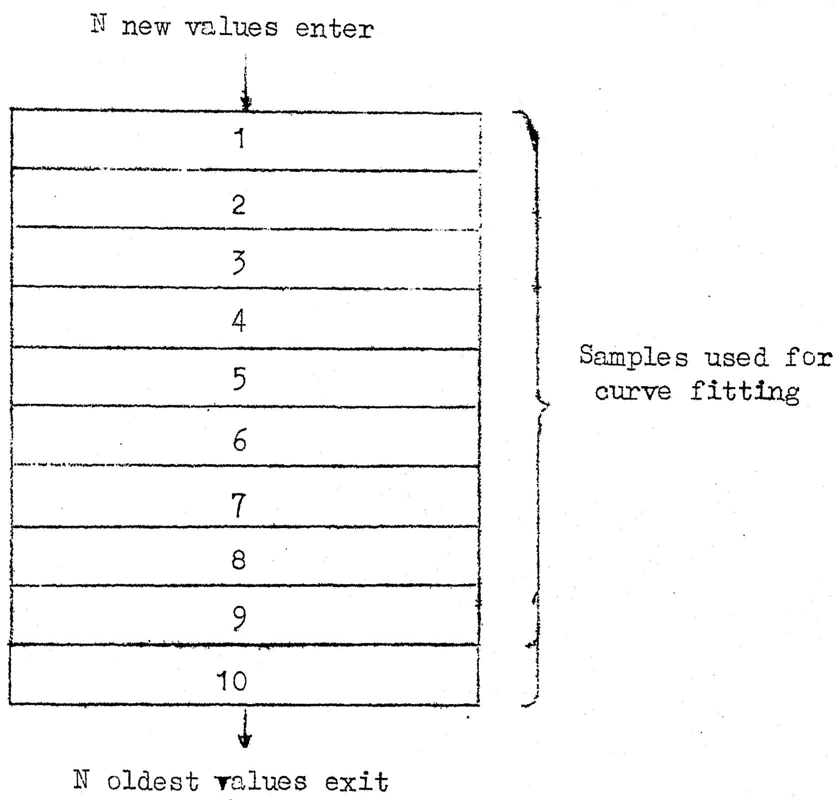


Fig. 3.4: Modified set of data for curve fitting

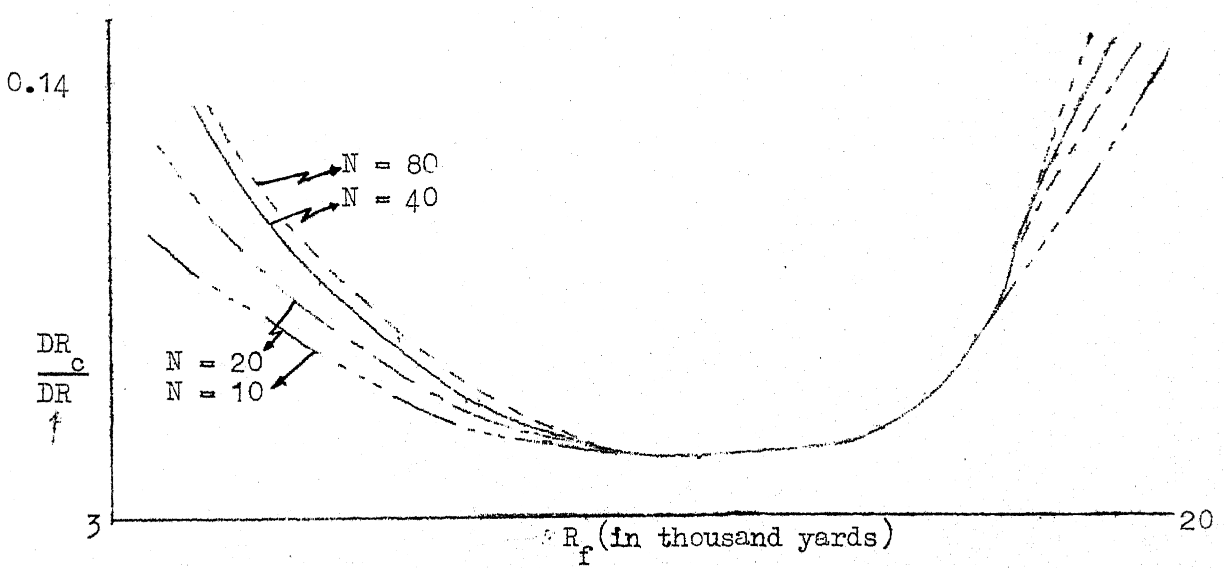


Fig. 3.5 (a) Variation of F_r with N

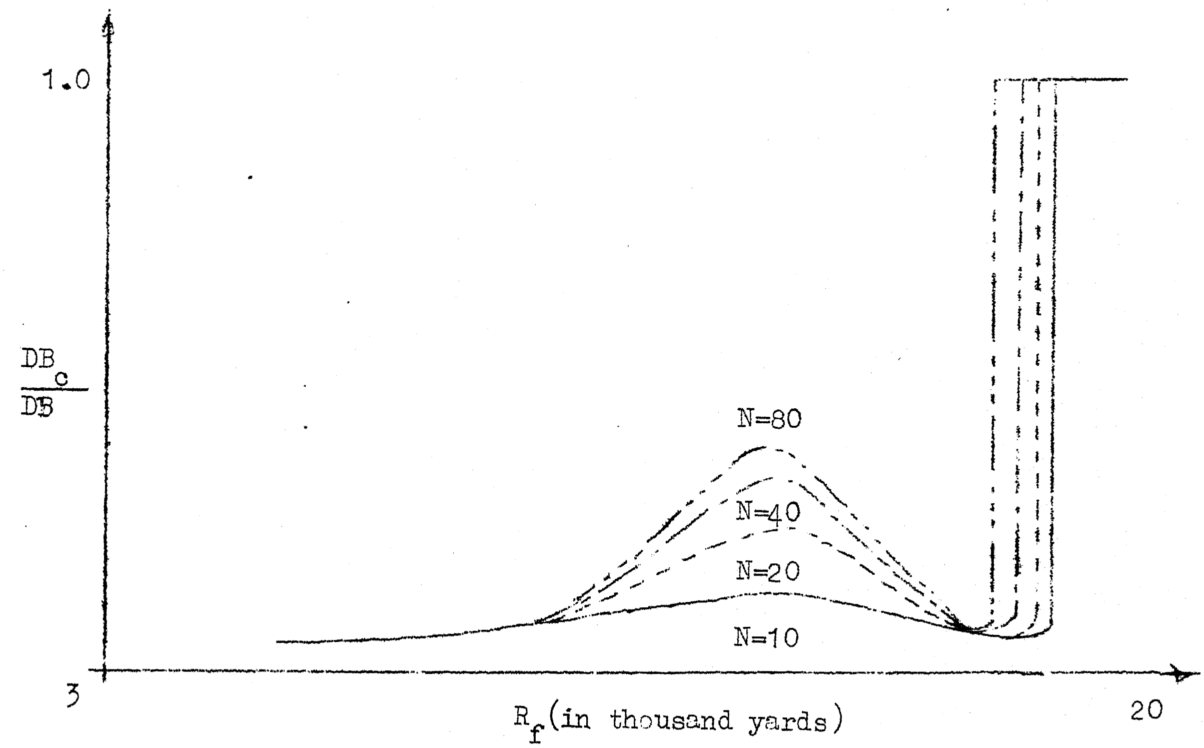


Fig.3.5(b)

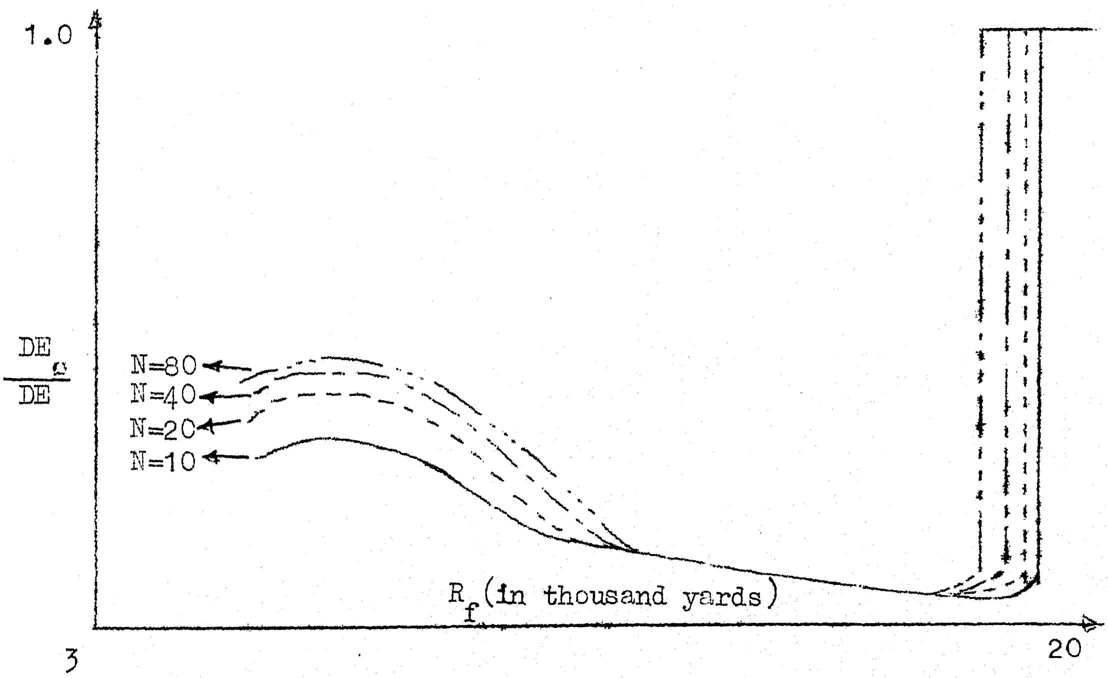


Fig. 3.5(c)

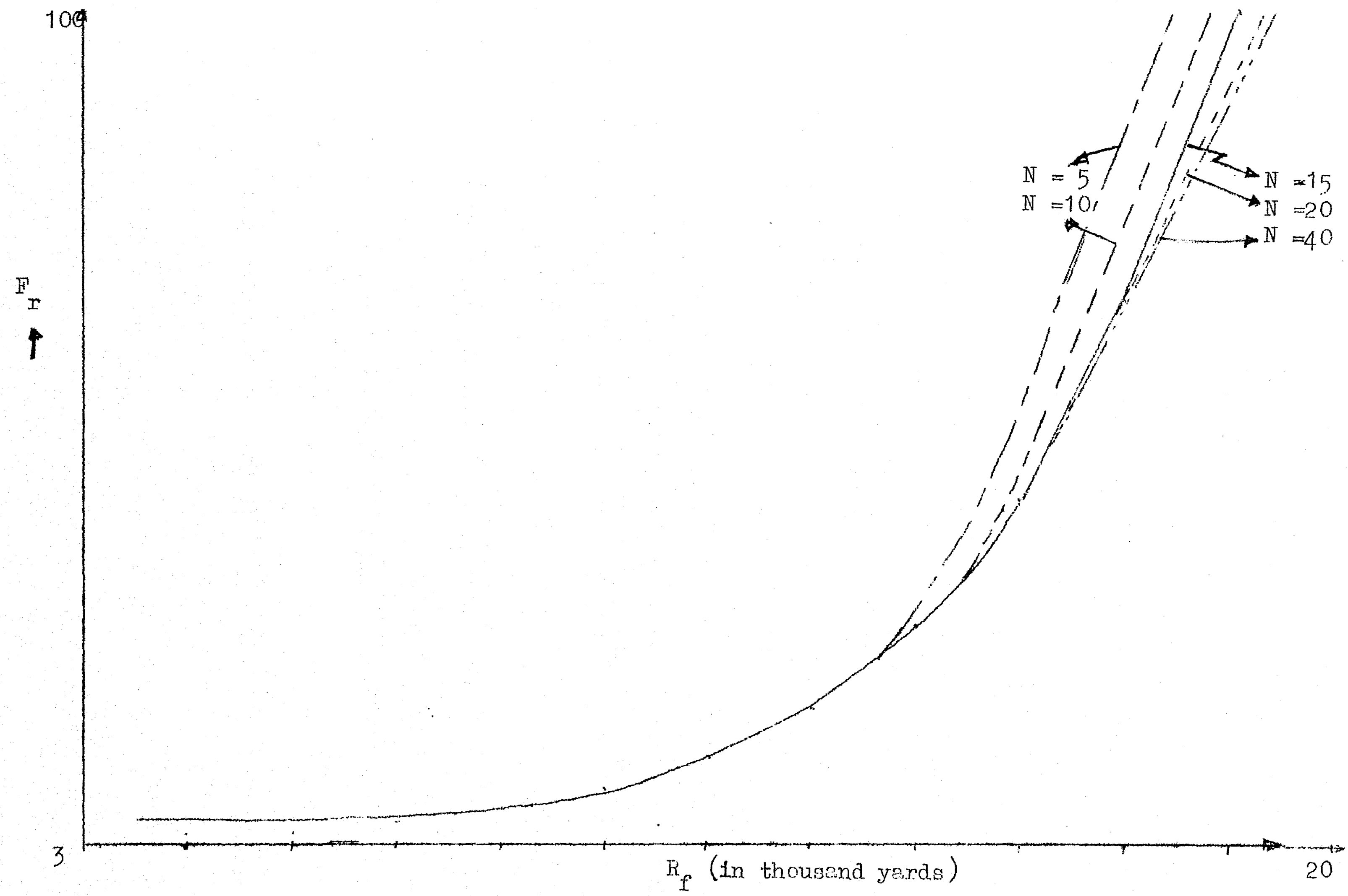


Fig. 3.6: Variation of performance factor with N

So far it has been assumed that the curve fitting routine is entered and the correction function coefficients are determined with ten samples of error values. This needs justification. The curve fitting has been tried with samples like 5, 10, 15, 20, 40 and 80 and the improvement factor in each case has been studied. The improvement factor for various number of samples is shown in Fig. 3.6. From this, it is seen that having more than 10 samples has very little effect on the improvement factor and it only adds to the complexity. Hence having only 10 samples for curve fitting is justifiable.

3.5 A Critical Analysis

In the sequal certain important aspects such as the cost of implementation of this technique and its complexities, are considered. For these discussions, a special purpose computer for the Weapon Control System is assumed to be available. In Chapter 4 the description of such a Special Purpose Computer will be given.

3.51 Time Delay in Determining the Correction Function Coefficients

The correction function coefficients are available only when the aircraft is at a range of about 20000 yards, though it was detected at a range of 30000 yards. That is, when the aircraft is detected at 30000 yards range, the present range is 30000 yards. For this, the predicted future range will be around 20000 yards when the aircraft speed is about 600 knots. Hence only when the aircraft comes within 20000 yards range, the first error value can be determined. As the curve fitting is done

when ten such samples of error values are available, the aircraft will be within 20000 yards range when the correction function coefficients are determined. This much time delay may not be acceptable. To minimize this delay the correction function coefficients are to be obtained much earlier for which suitable methods should be provided.

One method of getting over this difficulty is to assume a hypothetical Shell velocity which is much higher than the actual Shell velocity. A Shell velocity of ten times the initial value has been assumed in the beginning, to get the correction function coefficients earlier. Now, if the present range is 30000 yards, the predicted future range would be around 29000 yards. Thus the ten samples of error values required for curve fitting can be obtained when the aircraft is at a range of 29000 yards itself and the correction function coefficients can be determined. Once these coefficients are determined, future computations are done with the actual Shell velocity.

Now we have to see whether the use of these correction coefficients determined with the hypothetical Shell velocity, when applied to the computations with actual Shell velocity improve the performance of the system. After analysing the improvement factor with this method for about ten typical cases, it has been concluded that the improvement in the performance of the predictor is only marginal. The improvement factor is not more than 2. Hence it is decided to use these correction function coefficients obtained in this method till the correction coefficients

can be determined with actual Shell velocity. For the case discussed just now, for the period, the aircraft moves from 29000 yards to 20000 yards, the correction function coefficients determined with the hypothetical Shell velocity will be used and when the aircraft is within 20000 yards range, correction function coefficients determined with actual Shell velocity will be used.

3.52 Cost of Implementation of the Technique Vs Improvement in Performance

When this correction technique is to be implemented in the existing system, this adds to the complexity of the system as given below:

1) Additional storage space is required for the program.
2) Additional data storage for storing the predicted future position and samples of error in prediction.

This also increases the execution time of the programme.

This simple technique is independant of the system and is applicable for any system using a predictor. Also even when the aircraft is out of firing range, i.e during the initial tracking period, correction function coefficients are determined so that when the aircraft enters the firing range, it will be possible to shoot down the aircraft with better accuracy.

As it has been seen earlier, the improvement factors $F_r(x)$, $F_b(x)$ and $F_e(x)$ are of the order of 10. Hence if a weapon control system having 5% probability of 'kill' is chosen for the implementation of the technique developed, the probability of 'kill' of the modified system will be above 40%. In general,

$$P_m = 1-(1 - P_o)^n \quad * \qquad [3.35]$$

where P_m : probability of 'kill' of the modified system

P_o : Probability of 'kill' of the original system

and n : Improvement factor of the correction technique

Such an improvement in the system performance is of great advantage in Weapon Control Systems. Even an improvement factor of 2 will justify the cost of implementing the correction technique in Weapon Control Systems.

---

* Derivation of equation (3). is shown below with the old system.

Probability of aircraft escaping $= 1 - P_o$

With the modified system having an improvement factor, n.

probability of aircraft escaping $= (1 - P_o)^n$

$\therefore$ Probability of 'kill' of the modified system, $P_m = 1-(1-P$

PART - II

A SPECIAL PURPOSE COMPUTER

CHAPTER 4

A SPECIAL PURPOSE COMPUTER FOR WEAPON CONTROL

In this chapter a special purpose Digital Computer (used with a weapon control system) will be discussed. In military applications, there are stringent conditions on the factors like power requirements, space requirements, reliability and temperature range in which the system has to operate in addition to techniques for the removal of the dissipated heat.

Digital systems using vacuum tubes had a number of problems like heat dissipation,size and reliability. The mean time between failures (MTBF) of such systems were only few hours. Developments in the semiconductor devices and the Integrated circuits (IC) technology have made it feasible to design smaller and less costlier but more reliable systems. The high packing density of ICs, has further reduced the power consumption of such special purpose digital systems. In view of this, the system described below will have Integrated Circuits design.

4.1 Use of the ICs in Digital Systems

When small units like gates and units such as flip flops which use less than 9 gates are formed on a single monolithic chip, it is called Small Scale Integration (SSI)[3]. Whereas in Medium Scale Integration (MSI), logic sub assemblies such as counters, shift registers, decoders, memory elements, full adders and look ahead carry generators which use less than 100 gates are formed on a single monolithic chip. More complex circuits which need more than 100 gates such as Analog to Digital (A/D) converter and

Digital to Analog (D/A) converters are made in a single monolithic chip in Large Scale Integration (LSI). Putting larger sub-systems on a single chip has the following advantages:

1) Reduced Space and weight
2) Higher reliability due to reduction in external inter-connections.
3) Easy maintenance due to the minimization of test points.
4) Less cost per unit.

Designing systems with IC sub-assemblies is much simpler. Further there is reduction in inventory and spare parts stocking. The failure rate of individual gates are of the order of $3x10^{-7}$ per hour. Hence the MTBF for a computer using IC sub assemblies will be of the order of few thousand hours. Hence the computer described here will use ICs.

In Chapter 3, a clocking rate of 100 msec has been considered to be sufficient for the weapon control system under discussion. However a provision for flexibility in the system is required to deal with the improved speed in future aircrafts and to perform additional functions as they may be required. Hence it is better to have a speed improvement factor of 10. Thus the special purpose computer under discussion should be able to solve the fire control problem within an optimum time of 10 msec. Due to the practical considerations, the system should have minimum power dissipation and the cost should be reasonable.

Integrated Circuits use many different logic types. Some of them are given below:

1) Direct coupled transistor logic - DCTL

2) Resistor - transistor logic - RTL

3) Resistor - Capacitor-transistor logic - RCTL

4) Diode - transistor logic - DTL

5) Transistor-transistor logic - TTL

6) Emitter coupled logic - ECL

Of these, ECL is the fastest circuit, having switching time of less than 5 nsec with power consumption of about 50 mw per gate. Whereas TTL family has a wide variety of gates for different applications. The standard series TTL gates have typical propagation delay of 9 to 13 nsec and power dissipation of about 10mw per gate. High speed TTL series have switching speed of about 5 nsec but the power dissipation is about 35 mw per gate. Low power TTL series gates have power dissipation as low as 1 mw per gate but the propagation delay is about 33 nsec. Of all the IC logic types, TTL gives the highest performance to cost ratio[3] and among the various TTL series, the low power TTL series has the best speed-power product. But the low power TTL series are many times costlier than standard series TTL. Hence the low power TTL series units or sub assemblies can be used in systems where the cost factor is not very important compared to power requirements. In the system under consideration, standard series TTL circuits will be used as the increase in power requirements due to this, is reasonable. As the computer has to operate in the temperature range of -55°C to +125°C, logic sub assemblies designed for the operation in this temperature range, such as the 5400 series (of the Texas Instruments Inc.), will be used. The flow diagram of the proposed system is given in Fig.4.1.

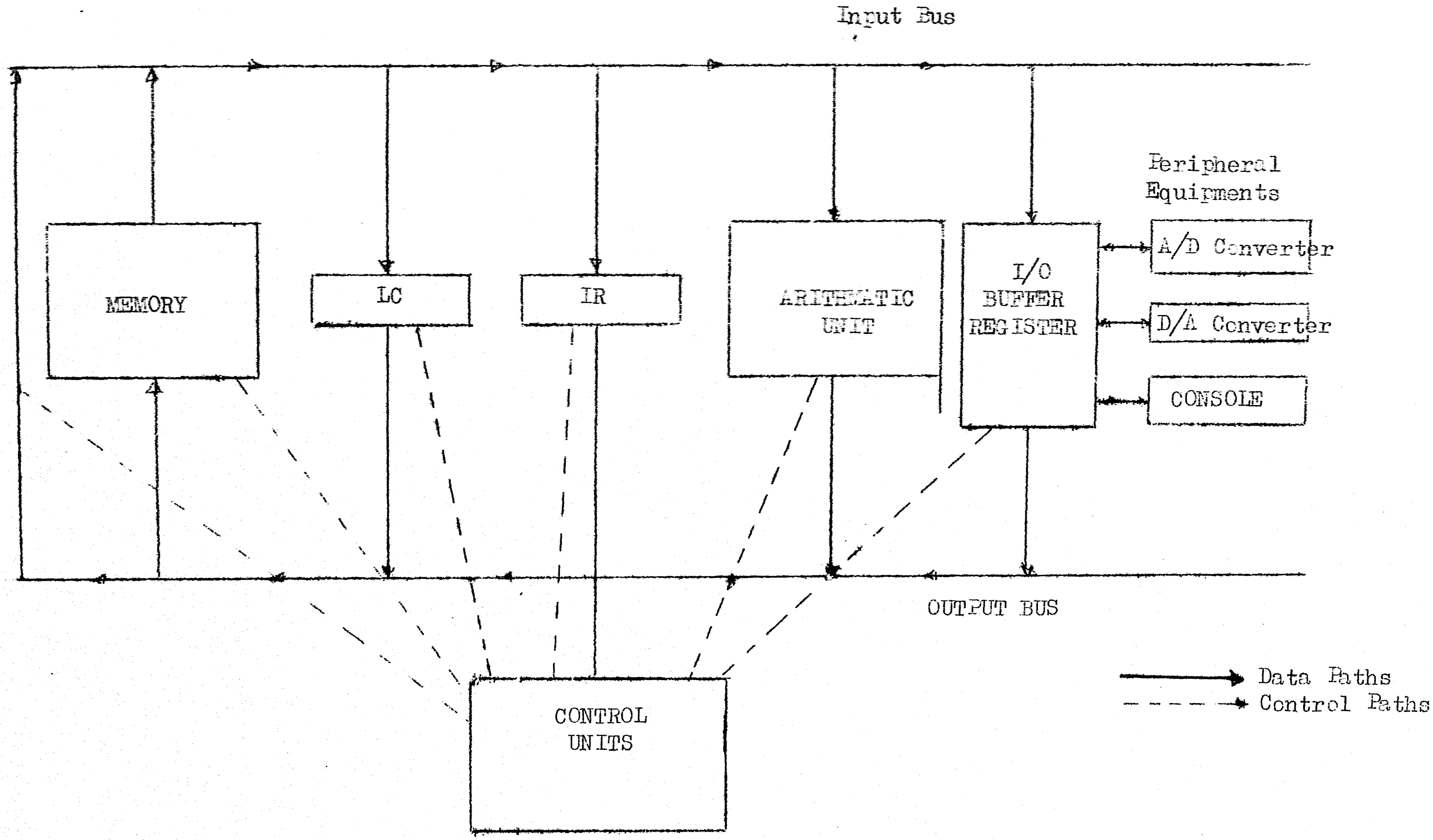


Fig. 4.1: Flow Diagram of the computer organisation

4.2 Selection of the Instruction Set

Considering the various calculations involved in the solution of the fire control problem, a basic set of instructions given below are proposed in the first round.

1) Loading a register from memory
2) Addition
3) Subtraction
4) Multiplication
5) Division
6) Storing the contents of a register in memory
7) Test for positivity
8) Complementation
9) Sampling the input
10) Outputting (the computed results)

Further provisions are to be made for generating functions, such as:

1) SINE (X)
2) COSINE (X)
3) ARCSINE (X)
4) SQUARE ROOT (X)

This can be done in two ways. The first is a software approach where subroutines will be stored to realise these functions. The second approach is to store these functions in microprogrammed memory and realise them hardwarewise[4]. First, the software approach will be tried to see whether this will suit the requirements of the system under discussion. Having

microprogrammed memory adds to the complexity of the system. Hence the hardware approach will be considered only if the software approach is not suitable for the system under consideration.

The following mnemonics are chosen for further discussion:

Mnemonics:	Operation
1) LD:	Load the Accumulator Register with the contents of the memory location addressed.
2) AD:	Add the contents of the addressed memory location to the contents of the Accumulator (Acc) and leave the result in the Acc.
3) SB:	Subtract the contents of the addressed memory location from the contents of the Acc and leave the result in the Acc.
4) MU:	Multiply the contents of the Acc by the contents of the addressed memory location and leave the result in the Acc.
5) DV:	Divide the contents of the Acc by the contents of the addressed memory location and leave the result in the Acc.
6) ST:	Store the contents of the Acc in the memory location addressed.
7) EN:	Execute the addressed subroutine with the contents of the Acc as the argument.
8) CS:	Complement the contents of the Acc.
9) TP:	If contents of the accumulator is positive transfer to the location whose displacement is given in bits 4 through 11 of IR (Instruction Register), else continue.

10) <u>RN:</u> Return to the routine from where a subroutine was called.

11) <u>SP:</u> Sample the input indicated by bits 8 through 11 of IR and store it in the Acc.

12) <u>OU:</u> Output the contents of the Acc, through D/A converter, to the output terminal indicated by bits 9 through 11 of IR.

4.3 Memory Requirements

With the instruction set given above, the fire control problem is programmed and is given in Appendix-II. From this, the following can be observed:

1) Memory required for main program = 276 locations
2) Memory required for SIN/COS Subroutine = 45 locations
3) Memory required for SQRT Subroutine = 21 locations
4) Memory required for ARSIN Subroutine = 37 locations

Thus total memory requirements of the program = 379 locations.

This special purpose computer operates essentially on a fixed program. From Appendix II, the memory requirements for data storage is 91. Hence 512 words of memory will be sufficient for the computer.

4.31 Data Format:

The ratio of the largest number to the smallest number to be handled by the computer under discussion is about 10^{13}. If fixed point arithmetic is to be used, about 44 bits word would be required to store the data. Hence 48 bits word length memory would be sufficient.

Whereas if the 48 bit word length is considered uneconomical, floating point arithmetic can be used. In floating point arithmetic a six bit signed exponent can handle numbers in the range of $2^{\pm 31}$ which is sufficient for the system under consideration. Further, the allowable error in the system is ± 3 yards (see section 2.1). This is 0.01 percent of 30000 yards which is the maximum range of interest. A choice for the word length as 24 bits with six bits for exponent and 18 bits for signed mantissa would satisfy such requirements. Thus the use of floating point arithmetic would require ninety one 24-bit word length memory locations. This saves ninety one 24-bit word length memory locations. But the use of floating point arithmetic adds to the complexity of the system. Further, provision for converting floating point number into fixed point representation and vice versa are to be provided. Considering these, fixed point arithmetic will be chosen for the computer, allowing ~~allowance~~ for tolerance of minor error in calculation.

4.32 Instruction format

The total memory requirement is less than 500 words. Hence 9 bits are required for the address part of the instruction. For the twelve operation codes (opcodes),four bits will be required and the instruction word length becomes 13 bits. A different organization is considered to use 12 bit words for storing the instructions. This will allow only 3 bits for opcode. With three bits, eight combinations are possible. As some of the opcodes do not need an address part, the eighth combination, namely, 111 is used to indicate such opcodes. The opcode table is shown below:

1)	LD	100
2)	AD	000
3)	SB	001
4)	MU	010
5)	DV	011
6)	ST	101
7)	EN	110
8)	TP	1111
9)	SP	111000
10)	OU	111001
11)	CS	111010
12)	RN	111011

As it is seen from the program given in Appendix II, the displacement for the conditional transfer instruction is always less than hundred locations. Hence the signed displacement for this instruction will need 8 bits namely bits 4 through 11 of I.R. Hence the four bit opcode 1111 is chosen for this instruction. Hence the first four bits of the remaining four opcodes, namely, SP, OU, CS and RN should be 1110. Hence their opcodes are chosen to have 6 bits as indicated in the opcode table. Further, instruction SP will use bits 8 through 11 to indicate the input terminal to be chosen and OU instruction will use bits 9 through 11 to indicate the output terminal to be chosen.

4.4 Input/Output Units

The computer under discussion will be executing a fixed program. Once the program is entered into the computer, it requires no modification.

In order to minimize the complexity of the computer it is decided not to have complex Input/Output devices. However an operator's console is considered essential. The program can be stored in magnetic cards. The data can conveniently be stored in IC memory.

The computer will have Analog/Digital and Digital/Analog converters. These will be used in multiplexer mode as there are thirteen inputs to be sampled and six computed results are to be given as output as indicated below:

Input 1	DT
Input 2	$\dot{R}$
Input 3	$\dot{S}$
Input 4	$\dot{B}_1$
Input 5	R_p
Input 6	S_p
Input 7	T_s
Input 8	S
Input 9	DTD
Input 10	E_{sp}
Input 11	B_{wr}
Input 12	WV
Input 13	E_q'
Output 1	T_g
Output 2	E_g
Output 3	(t_z+DT)
Output 4	R_{gz}
Output 5	B_{1d}
Output 6	S_d

These samplings are done in time interval of 100 msec as mentioned in chapter 2. Provision will be made for raising an alarm if either of the two converter outputs remain constant for an amount of preset time, say, for more than 5 msec. This would indicate a possible failure of the system.

4.5 Arithmetic Units

The following arithmetic units are required for this computer:

1) One 48-bit fixed point full adder
2) One 48-bit fixed point multiplier
3) One 48-bit fixed point divider.

On a single IC chip, fixed point adders are available. These circuits are capable of performing parallel addition and subtraction. Using repeated addition or subtraction combined with shifting right or left, multiplication and division algorithms can be realized[5].

The following set of registers will be required in the Central Processing Unit (CPU).

1) Two 48-bit Accumulator Registers, Acc
2) One 48 bit Multiplier Quotient Register, MQ.
3) One 12 bit Instruction Register, IR.
4) One 9 bit Location Counter, LC.
5) One 9 bit Register to store the contents of LC while calling a subroutine, 'C' Register.
6) One 48 bit Register to hold the addend during addition and subtraction, 'B' Register.

7) One 9-bit Memory Address Register, MAR

8) One 48-bit Memory Buffer Register, MBR

9) One 48-bit Input/Output Buffer Register, I/O Buffer

4.6 <u>Control Circuits</u>

There are two basic cycles in the computer operation. They are:

1) Instruction fetch cycle or I cycle

2) Execution cycle or E cycle

During the I-cycle, an instruction is read from the memory and brought to the IR. This involves the following sequence of operations.

T1: Shift the contents of the LC to MAR

T2: Give a read pulse and wait till the reading is complete.

T3: Shift the contents of MBR to IR.

The circuit shown in Fig. 4.2a will generate the timing sequence pulses for instruction fetch. The timing pulses are shown in Fig.4.2b. An instruction fetch is initiated under the following conditions:

1) Either the start signal is 'ON' or the instruction complete signal occurs.

2) Stop or Halt signal is 'OFF'.

3) The above two conditions coincide with a clock pulse.

The following assumptions are made and justified:

1) 'n' is the number of basic units of time or clock time, the I cycle takes.

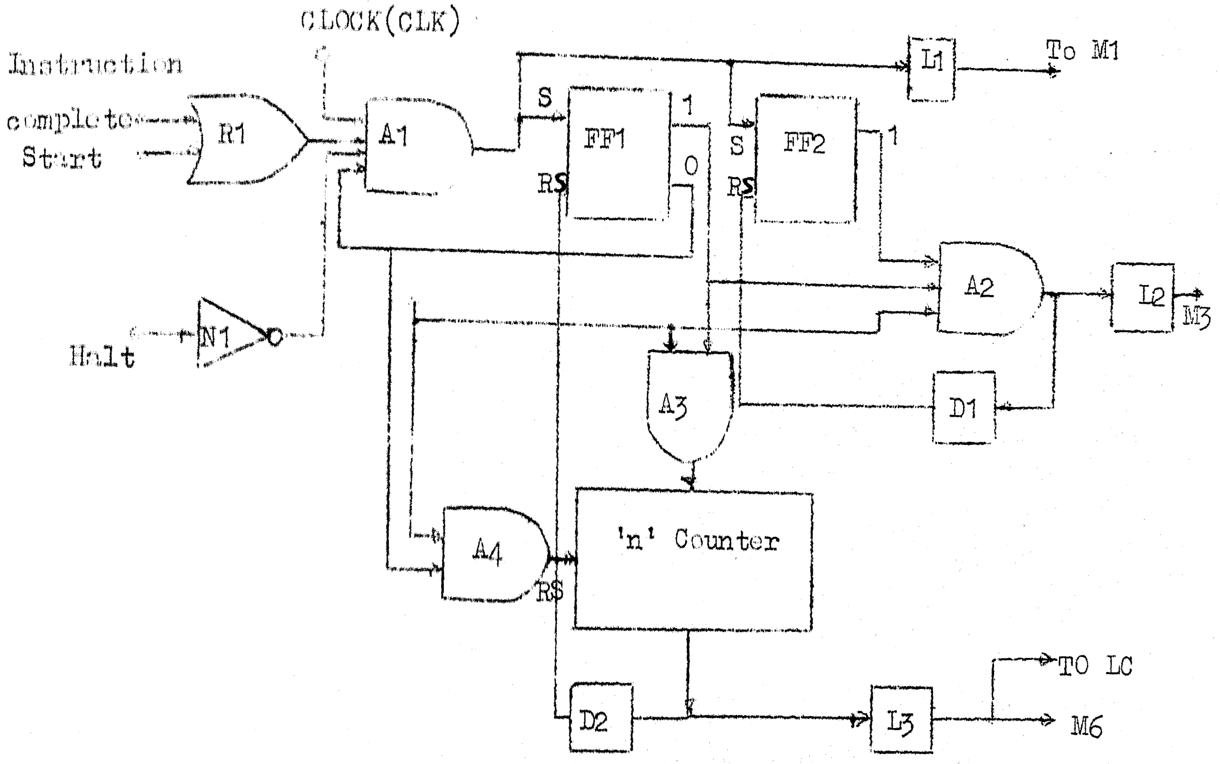


Fig. 4.2(a): Instruction Fetch Cycle

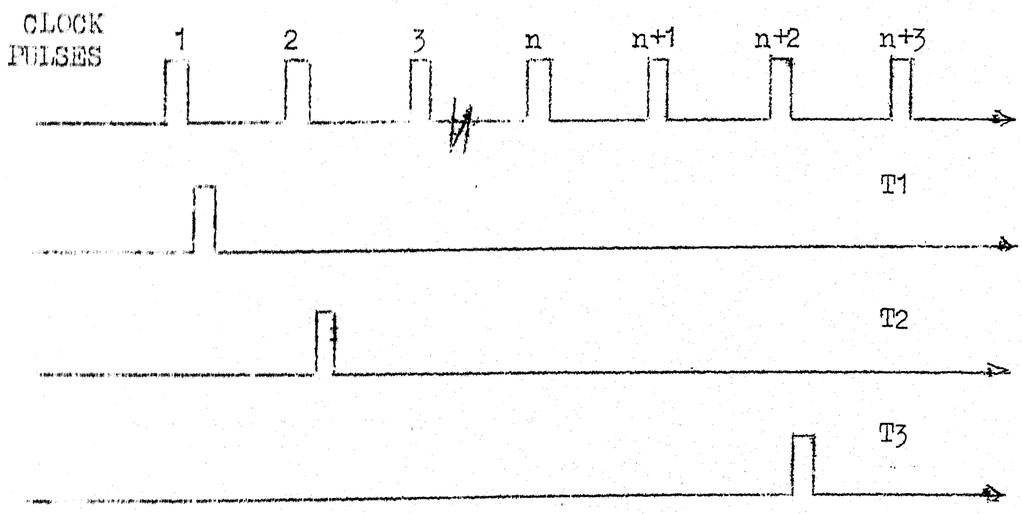


Fig. 4.2(b): Timing pulses for Instruction Fetch.

2) The time between two clock pulses is greater than the rise time of the flip flop, FF1. This is true since the clock time is 150 nsec as chosen in a later section whereas the rise time of the flip flop is around 20 nsec.

3) The clock pulse duration is less than the flip flop rise time. Hence the clock pulse duration is chosen to be less than 20 nsec.

4) Flip flop rise time is greater than the gate delay time. This is true as the gate delay is about 10 nsec.

5) Start pulse coincides with a clock pulse. This is true as the clock time is 150 nsec and the start pulse duration would be more than this.

On completion of the I-cycle, the E-cycle will commence. During E-cycle, first the opcode is decoded with the circuit shown in Fig.4.3. The sequence of micro operations to realize various opcodes are as shown below:

1) <u>LD-100:</u> Load accumulator with the contents of addressed memory location.

   i) Shift the contents of the address part of IR to MAR.

   ii) Give the read pulse and wait till the reading is over.

   iii) Shift the contents of MBR to Acc.

   iv) Switch the 'operation complete' signal 'ON'.

2) <u>AD-000:</u> Add the contents of the addressed memory location to the Acc.

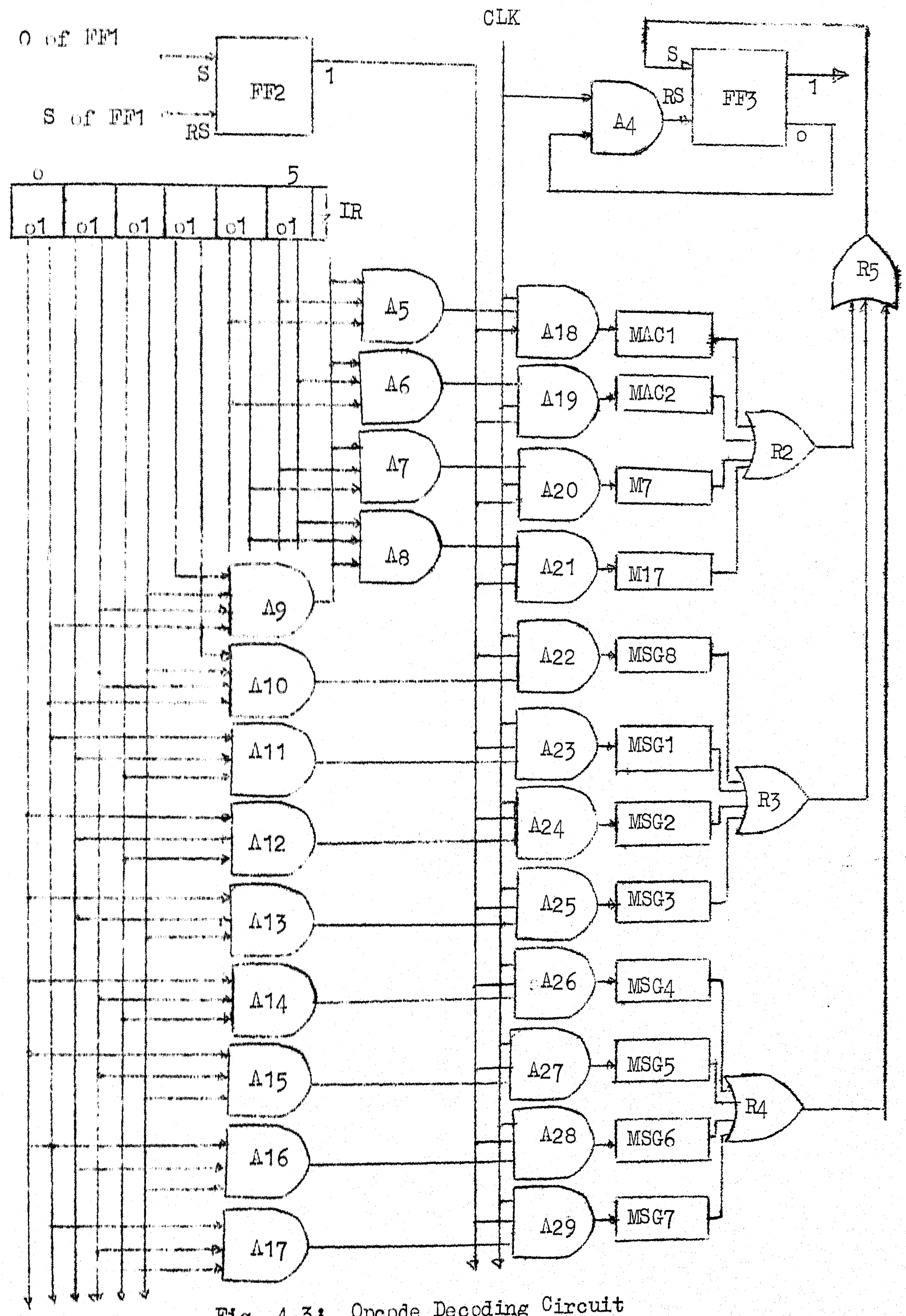


Fig. 4.3: Opcode Decoding Circuit

i) Shift contents of address part of IR to MAR.

ii) Give a read pulse and wait till the reading is complete.

iii) Shift the contents of MBR to 'B' register.

iv) Connect the Acc and 'B' register to the hardware adder and start Addition Cycle.

v) At the end of addition cycle switch the operation complete signal 'ON'.

3) <u>SB-001</u>: Subtract the contents of the addressed memory location from the contents of the Acc.

i) Shift address part of IR to MAR.

ii) Give a read pulse and wait till the reading is complete.

iii) Shift the contents of MBR to 'B' register.

iv) 1's complement the contents of 'B' register.

v) Connect the 'B' and Acc registers to the adder and start addition cycle.

vi) When the addition is complete, switch the operation complete signal "ON".

4) <u>MU-010</u>: Multiply the contents of the Acc by the contents of the addressed memory location:

i) Shift address part of IR to MAR.

ii) Give a read pulse and wait till the reading is complete.

iii) Shift the contents of MBR to 'B' register.

iv) Connect the 'B' and Acc registers to the Multiplier and start multiplication.

v) When the multiplication is complete, switch the operation complete signal 'ON'.

5) <u>DV-011</u>: Divide the contents of the Acc by the contents of the addressed memory location.

i) Shift address part of IR to MAR.

ii) Give a read pulse and wait till the reading is complete.

iii) Shift the contents of MBR to 'B' register.

iv) Connect the 'B' and Acc registers to the Divider and start division.

v) When the division is complete, switch the operation complete signal 'ON'.

6) <u>ST-101</u>: Store the contents of the Acc.

i) Shift the address part of IR to MAR.

ii) Shift contents of Acc to MBR.

iii) Give a write pulse and wait till the writing is complete.

iv) Switch the operation complete signal 'ON'.

7) <u>EN-110</u>: Execute the called subroutine with the contents of the Acc as argument.

i) Shift the LC contents to the 'C' register.

ii) Shift the address part of IR to LC.

iii) Switch the operation complete signal 'ON'.

8) <u>TP-1111</u>: Sign convention: Sign bit 0 -- positive
1 -- negative

Check the sign bit of Acc. If it is 1 give end of operation signal.

It the sign bit is 0,

i) Add the contents of bits 4 through 11 of IR to LC.

ii) Switch the end of operation signal 'ON'.

9) <u>SP - 111000:</u> Sample the input referred

i) Select the input terminal number as given by bits 8 through 11 of IR and connect it to the A/D converter. When the A/D conversion is over, store the output of A/D converter in the Acc.

ii) Switch the end of operation signal 'ON'.

10) <u>OU: 111001:</u> Output the result referred.

i) Connect the Acc to the D/A converter. Connect the output of D/A converter to the output terminal indicated by bits 9 through 11 of IR.

ii) When the conversion is over switch the operation complete signal 'ON'.

11) <u>CS: 111010:</u> Change the sign of the contents of Acc.

i) 1's complement the contents of Acc.

ii) Switch the operation complete signal 'ON'.

12) <u>RN - 111011:</u> Return to the main routine

i) Shift contents of 'C' register to LC.

ii) Switch the operation complete signal 'ON'.

Thus, a sub set of the following set of micro-operations have been used in all the micro programs given above.

M1: Shift contents of LC to MAR

M2: Shift address part of IR to MAR.

M3: Give read pulse.

M4: Shift contents of MBR to Acc.

M5: Shift contents of MBR to 'B' register.

M6: Shift contents of MBR to IR.

M7: 1's complement Acc.

M8: 1's complement 'B' register.

M9: Addition cycle.

M10: Multiplication cycle.

M11: Division cycle.

M12: Shift contents of Acc to MBR.

M13: Give a write pulse

M14: Shift contents of LC to 'C' register.

M15: Shift address part of IR to LC.

M16: Add bits 4 through 11 of IR to contents of LC.

M17: Shift contents of 'C' register to LC.

In addition to these micro-operations, the following Macro-operations have been used:

MAC1: A/D conversion.

MAC2: D/A conversion.

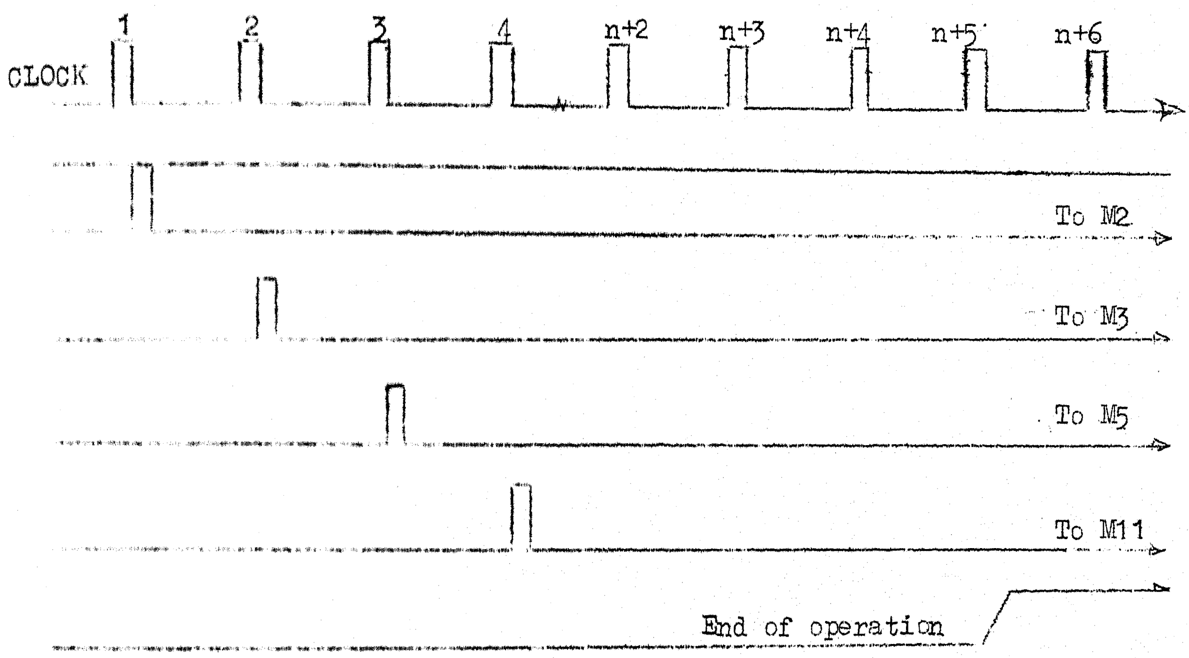
A gate level design of circuits for these Micro and Macro-operations are given in Appendix-III. Execution of each opcode involves selection of a set of these micro and macro-operations. This is indicated in table 4.1. For selecting these set of micro and macro operations either micro programmed memory or conventional circuits can be used. Micro/Macro sequence generators (MSG) use conventional circuits. When an opcode is decoded, the corresponding MSG gets a pulse and this MSG generates the necessary pulses to choose the micro and macro operations

Table 1.1

Selection of Micro / Macro Operations .

OPCODE \ SUBROUTINE	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	MAC1	MAC2
INSTRUCTION FETCH	✓					✓													
LD		✓	✓	✓															
AD		✓	✓		✓				✓										
SB		✓	✓		✓			✓	✓										
MU		✓	✓		✓					✓									
DV		✓	✓		✓						✓								
ST		✓										✓	✓						
EN														✓	✓				
TP																✓			
SP																		✓	
OU																			✓
CS							✓												
RN																	✓		

to be executed. A gate level design of the MSGs is given in Appendix IV. The timing pulses generated by MSG3 to execute the opcode DV, is given below, as an example:



4.7 Selection of Clock rate:

It is observed from the list of Micro-operations, that many of the micro-operations involve shifting of contents of one register to another register. The time required for such operations will be chosen as the basic time of the computer. This operation will involve typically two levels of gate delay and the time for a J-K flip flop to toggle. When J-K flip flops designed for a toggle rate of 20MHz are used as with standard series TTL gates, the total time delay would be about 100 nsec. A clock rate of 150 nsec will satisfy the requirement.

TABLE 4.2

No.	OPCODE / SUBROUTINE	No. of times executed	Time for one execution (μsec)	Total time (μsec)
1	LD	59	0.6	35.4
2	AD	35	1.0	35.0
3	SB	16	1.15	18.4
4	MU	86	1.6	137.6
5	DV	8	3.5	28.0
6	ST	102	0.45	45.9
7	EN	28	0.15	4.2
8	TP	8	0.45	3.6
9	SP	13	13.0	169.0
10	OU	6	13.0	78.0
11	CS	2	0.15	0.3
12	RN	--	0.15	---
13	SIN	9	36.4	327.6
14	COS	6	38.6	231.6
15	ARSIN	8	195.2	1561.6
16	SQRT	5	276.7	1383.5
			TOTAL TIME	4 msec.(Approx.)

4.8 Total time for execution of the program and power requirements:

The number of times each instruction is executed and the time for E-cycle of various instructions, are tabulated in table 4.2. Analog/Digital and Digital/Analog converters with 500 KHz parallel word rate are assumed[5]. The optimum time to solve the fire control problem is 10 msec as decided in Section 4.1. Within this 10msec, the following should be done:

1) Sample all 13 inputs with A/D converters.

2) Compute the output variables from the input variables.

3) Convert the output into Analog signals with D/A converters.

From table 4.2, it is seen that the execution time excluding I-cycle is approximately 4 msec. The total number of instruction fetch in one iteration is about 1600. Thus one I-cycle can take a maximum time of about 3.7 μsec. Conventional core memories with 2 μsec cycle time would very much serve the purpose.

4.9 Advantages of the proposed computer:

The special purpose computer described here has the following advantages when compared with one of the existing Electromechanical computers for solving the fire control problem.

1) Maintenance: The system described here, has logical sub-assemblies as basic unit. Hence maintenance becomes comparatively easy. Also this effects a reduction in inventory and spare parts stocking.

2) Reliability: As explained in section 4.1, the mean time between failure for the proposed system is of the order of few

thousand hours and hence the system is much more reliable than the existing system.

3) Speed: In the existing system, there is lag in each servo unit. Due to this, the entire system becomes sluggish. Whereas in the proposed new system, the time for computing the results is less than 10 msec. Even this delay can be taken care off by applying a correction term to the predicted time of flight of the Shell, t_f.

4) Space: The electromechanical system will require a big room to install all the units. But the proposed new system can be kept in a small compact unit (apart from its 512 word memory and operator's console).

5) Power consumption: In the electro-mechanical system using vacuum tubes, power dissipation and the removal of disspated heat pose a big problem. Whereas the proposed new system consumes less than 30 watts power as can be seen from table 4.3. This is less than 5 percent of the power requirements of the existing system. Further ICs can operate satisfactorily in the temperature range of -55°C to +125°C which is good enough for the weapon control system. Cooling the new system is very simple as the amount of heat to be removed is very small.

6) Operational procedure: Once the new system is installed, it needs very little modifications. Testing is comparatively simple. The new system does not need constant attention of

TABLE 4.3

Cost and Power Requirements

No.	Item	No. Reqd.	Power per unit mw	Total Power mw	Approx. cost $
1	8 bit, 32 word IC memory	9	240	2160	500.0
2	2 input, AND Gate	668	10	6680	101.87
3	3 input AND gate	25	10	250	6.84
4	4 input AND gate	13	10	130	5.32
5	5 input AND gate	12	10	120	9.12
6	2 input OR gate	4	10	40	0.76
7	3 input OR gate	2	10	20	0.76
8	4 input OR gate	5	10	50	2.28
9	5 input OR gate	1	10	10	0.76
10	1 out of 16 input selector	1	150	150	6.98
11	8 line Multiplexer	1	100	100	5.85
12	Memory Driver	1	50	50	3.08
13	R-S flip flop	27	40	1080	30.0
14	J-K flip flop	3	40	120	3.81
15	Latches	21	40	840	35.0
16	4 bit shift registers	6	50	300	21.6
17	4 bit Arithmetic logic Unit	12	100	1200	32.88
18	Look ahead carry generator	4	120	480	20.84
19	4 bit binary counters	11	130	1430	33.66
20	12 bit A/D Converter	4	300	1200	248.0
21	10 bit D/A converter	5	300	1500	282.5
22	Shift Registers- 16 bit	13	200	2600	52.56
			TOTAL	21430	1404.47

the operator. There is no need either to tune the system or to balance the amplifier output as required in the existing system.

7) Cost: The cost of the proposed system is expected to be around Rs.20,000, which is comparable to the cost of the existing system, due to the added advantages of the new system.

8) Mechanical parts: There is no mechanical moving parts in the proposed system. Whereas the existing system has wear and tear due to mechanical moving parts.

9) Other considerations: In the proposed system with extended memory, the error correction technique as developed in chapter 3 can be implemented to improve the performance of the weapon control system.

CHAPTER 5

CONCLUSIONS AND SCOPE FOR FURTHER WORK

A predictor correction technique has been developed and found to work satisfactorily for forty eight typically chosen cases, which include most of the possibilities that occur in real situations. They represent combinations of the following situations:

A1: Aircraft detected at long range.

A2: Aircraft detected at close range.

B1: Approaching Aircraft.

B2: Aircraft moving away.

C1: Faster aircraft.

C2: Slow aircraft.

D1: Low flying aircraft.

D2: Aircraft at high altitude and diving down.

D3: Aircraft at high altitude and not diving down.

E1: Calm atmosphere.

E2: Moderately high wind speed.

The special purpose computer described here has many advantage over the existing system like, better speed, easy maintenance, better reliability, simple operational procedure, lesser power consumption and requires lesser space (see section 4.9). A logical organization for this special purpose computer has been discussed. There is scope for further work in the following lines:

1) Sufficient data regarding the course an aircraft takes under various real time situations can be collected and stored. This can be used to account for the pilot's actions.

2) The logical design of the proposed computer can be tested by simulation on an existing Digital Computer.

3) The proposed computer can be fabricated and tested under the actual field conditions with a weapon control system.

4) The proposed computer can be modified to have extended memory and the predictor correction technique as developed in Chapter 3 can be implemented.

5) A modified version of the proposed computer with extended facilities can be situated in a centralised place and used to control many weapon systems.

APPENDIX I

LIST OF TYPICAL CASES ANALYSED

The following is the list of 48 typically chosen cases which were analysed:

No.	R_p (Yards)	S_p (Deg.)	T_s (Deg.)	X (Deg.)	Y (Deg.)	V (Knots)	WV (Knots)	B_{wr} (Deg.)
1	6623.8	2.0	343.6	180.0	0.0	300.0	8.5	195.7
2	7286.2	2.0	43.6	180.0	0.0	300.0	24.6	104.3
3	6876.9	2.0	173.8	180.0	0.0	600.0	18.8	39.3
4	5553.4	2.0	191.8	180.0	0.0	600.0	38.4	257.0
5	30000.0	2.0	123.7	0.0	0.0	300.0	11.9	293.9
6	30000.0	2.0	254.6	0.0	0.0	300.0	29.2	78.7
7	30000.0	2.0	158.3	0.0	0.0	600.0	11.8	168.3
8	30000.0	2.0	280.3	0.0	0.0	600.0	29.0	280.1
9	6956.2	8.0	229.2	180.0	0.0	300.0	19.6	138.1
10	4593.4	8.0	109.9	180.0	0.0	300.0	39.5	281.0
11	6886.4	8.0	72.8	180.0	0.0	600.0	5.6	246.5
12	4634.6	8.0	296.8	180.0	0.0	600.0	20.8	353.1
13	30000.0	8.0	348.1	0.0	0.0	300.0	9.2	320.6
14	30000.0	8.0	148.9	0.0	0.0	300.0	25.5	155.0
15	30000.0	8.0	331.8	0.0	0.0	600.0	12.4	352.4
16	30000.0	8.0	300.6	0.0	0.0	600.0	29.9	280.1
17	6500.0	8.0	13.5	180.0	-3.8	300.0	15.8	21.1
18	5521.6	8.0	45.8	180.0	-5.9	300.0	34.4	177.7

Contd...

No.	R_p (Yards)	S_p (Deg.)	T_s (Deg.)	X (Deg.)	Y (Deg.)	v (Knots)	WV (Knots)	B_{wr} (Deg.)
19	6812.6	8.0	25.6	180.0	-4.9	600.0	13.6	36.5
20.	3117.9	8.0	22.7	180.0	-5.6	600.0	31.5	133.7
21.	30000.0	8.0	340.9	0.0	-3.6	300.0	14.8	324.4
22	30000.0	8.0	160.0	0.0	-4.4	300.0	33.0	318.9
23	30000.0	8.0	37.0	0.0	-3.6	600.0	16.3	180.0
24	30000.0	8.0	286.8	0.0	-5.3	600.0	35.0	51.6
25	6075.4	2.0	143.6	180.0	0.0	300.0	13.0	133.6
26	4627.7	2.0	3.3	180.0	0.0	300.0	30.0	206.1
27	4835.5	2.0	158.0	180.0	0.0	600.0	17.9	45.2
28	5104.5	2.0	202.2	180.0	0.0	600.0	37.2	55.0
29	17103.5	2.0	127.1	0.0	0.0	300.0	9.3	186.9
30	19204.1	2.0	21.3	0.0	0.0	300.0	25.8	248.8
31	18701.0	2.0	155.3	0.0	0.0	600.0	13.4	10.7
32	15193.0	2.0	225.1	0.0	0.0	600.0	31.2	334.3

Contd ....

No.	R_p (Yards)	S_p (Deg.)	T_s (Deg.)	X (Deg.)	Y (Deg.)	v (Knots)	WV (Knots)	B_{wr} (Deg.)
33	4521.2	8.0	296.4	180.0	0.0	300.0	7.6	317.1
34	4724.9	8.0	254.7	180.0	0.0	300.0	23.4	141.3
35	6616.2	8.0	196.3	180.0	0.0	600.0	18.0	189.5
36	7517.1	8.0	32.0	180.0	0.0	600.0	37.4	336.5
37	19435.0	8.0	49.7	0.0	0.0	300.0	14.6	151.5
38	17828.5	8.0	68.7	0.0	0.0	300.0	32.7	208.9
39	18569.2	8.0	234.0	0.0	0.0	600.0	11.1	115.1
40	15772.0	8.0	28.9	0.0	0.0	600.0	28.1	70.0
41	6722.8	8.0	219.2	180.0	-3.6	300.0	6.5	0.8
42	6088.1	8.0	265.4	180.0	-3.4	300.0	22.0	308.4
43	7406.2	8.0	334.0	180.0	-4.5	600.0	5.1	221.7
44	5932.7	8.0	30.3	180.0	-4.3	600.0	20.1	249.1
45	16035.8	8.0	64.4	0.0	-3.7	300.0	13.0	346.4
46	19083.5	8.0	300.6	0.0	-4.4	300.0	30.7	35.5
47	15681.7	8.0	190.4	0.0	-5.1	600.0	17.7	313.4
48	16735.5	8.0	162.6	0.0	-4.6	600.0	36.9	106.4

APPENDIX - II

PROGRAM TO SOLVE THE FIRE CONTROL PROBLEM

To solve the fire control problem with the special purpose computer described in Chapter 4, the following program is used:

MAIN PROGRAM

SP	INPUT 1
ST	DT
SP	INPUT 2
ST	$\ddot{R}$
SP	INPUT 3
ST	$\ddot{S}$
SP	INPUT 4
ST	$\dot{R}_1$
SP	INPUT 5
ST	R_p
SP	INPUT 6
ST	$\dot{S}_p$
SP	INPUT 7
ST	T_s
SP	INPUT 8
ST	S
SP	INPUT 9
ST	DTD

(Continued in next column)

SP	INPUT 10
ST	E_{sp}
SP	INPUT 11
ST	B_{wr}
SP	INPUT 12
ST	WV
SP	INPUT 13
ST	E'_q
LD	$\ddot{R}$
MU	dt
AD	$\dot{R}$
ST	$\dot{R}$
MU	dt
AD	R_p
ST	R_p
LD	$\ddot{S}$
MU	dt
AD	$\dot{S}$

Contd,...

ST	$\dot{S}$
LD	$\ddot{B}_1$
MU	dt
AD	$\dot{B}_1$
ST	$\dot{B}_1$
LD	S_p
EN	SIN
ST	Sin S_p
LD	S_p
EN	COS
ST	Cos S_p
LD	S
EN	SIN
ST	Sin S
MU	S
ST	Acc2
LD	S
EN	COS
ST	COS S
MU	$\dot{B}_1$
AD	Acc2
ST	$\dot{B}_{1d}$
LD	$\dot{B}_1$

(Continued in the next column)

MU	Sin S
ST	Acc2
LD	$\dot{S}$
MU	Cos S
SB	Acc2
ST	$\dot{S}_d$
LD	$\dot{S}$
MU	$\dot{S}$
ST	Acc2
LD	$\dot{B}_1$
MU	$\dot{B}_1$
AD	Acc2
EM	SQRT
ST	w
LD	$\dot{B}_1$
MU	Sin S_p
DV	Cos S_p
ST	$\dot{C}$
MU	$\dot{B}_1$
ST	$\dot{B}_1\dot{C}$
LD	$\dot{R}$
MU	CT1
MU	w

Contd....

	DV	R_p
	ST	$\dot{w}$
	MU	Sin C
	ST	Acc2
	LD	$\dot{S}$
	MU	$\dot{C}$
	AD	Acc2
	ST	$\ddot{R}_1$
	LD	$\dot{w}$
	MU	CosC
	SB	$\dot{R}_1 C$
	ST	$\ddot{S}$
	LD	R_p
	MU	w
	ST	$R_p w$
	MU	w
	ST	$\ddot{R}$
	LD	S_f
	EN	COS
	ST	Cos S_f
	LD	S_f
	EN	SIN
	ST	Sin S_f
	LD	WV

(Continued in the next column)

	MU	CT2
	DV	CT3
	ST	WVC
LA	LD	R_f
	ST	R_{fo}
	LD	$\dot{R}$
	MU	t_f
	AD	R_p
	ST	Acc2
	MU	Acc2
	ST	Acc2
	LD	$R_p w$
	MU	t_f
	ST	R_fSin D
	MU	R_fSin D
	AD	Acc2
	EN	SQRT
	ST	R_f
	LD	R_fSin D
	DV	R_f
	ST	Sin D
	EN	ARSIN
	ST	D
	MU	Cos C
	AD	S_p

Contd...

	ST	S_f
	LD	CT4
	MU	R_f
	AD	CT5
	MU	S_f
	ST	Acc2
	LD	Bwr
	SB	T_g
	SB	DTD
	ST	Bwg
	EN	SIN
	ST	Sin Bwg
	LD	Bwg
	EN	COS
	ST	Cos Bwg
	MU	WV
	MU	R_f
	MU	CT6
	MU	Cos S_f
	ST	KW
	AD	Acc2
	MU	t_f
	AD	R_{fo}

(continued in the next column)

	ST	Acc2
	MU	CT7
	EN	ARSIN
	MU	CT8
	ST	TE
	EN	SIN
	MU	CT9
	ST	t_f
	LD	R_f
	SB	R_{fo}
	TP	PS
	CS	...
PS	SB	CT10
	TP	LA
	LD	CT11
	MU	Cos S_f
	DV	Acc2
	AD	TE
	ST	(TE+DIP)
	MU	CT12
	ST	DRIFT
LB	LD	t_z
	ST	t_{zo}

Contd...

	AD	DT
	ST	(t_z+DT)
	MU	$\dot{R}$
	AD	R_p
	ST	Acc2
	MU	Acc2
	ST	Acc2
	LD	$R_p w$
	MU	(t_z+DT)
	ST	R_z Sin D_z
	MU	R_z Sin D_z
	AD	Acc2
	EN	SQRT
	ST	ACC2
	LD	KW
	MU	t_z
	AD	Acc2
	ST	R_{gz}
	MU	CT7
	EN	ARSIN
	MU	CT8
	EN	SIN
	MU	CT9
	ST	t_z

(Continued in the next column)

	SB	t_{zo}
	TP	RS
	CS	..
RS	SB	CT13
	TP	LB
	LD	E_{sp}
	EN	COS
	ST	Cos E_{sp}
	LD	Sin C
	MU	Sin S
	ST	Acc2
	LD	Cos C
	MU	Cos S
	SB	Acc2
	MU	Sin D
	MU	Cos E_{sp}
	ST	Acc2
	EN	ARSIN
	ST	E_{sf}
	LD	R_f
	MU	WVC
	ST	RWC
	MU	Cos B_{wg}
	MU	CT14

Contd....

MU	Sin S_f
ST	Acc2
LD	(TE+DIP)
MU	Cos S_f
AD	Acc2
ST	cDv
LD	RWC
MU	Sin B_{wg}
MU	CT15
AD	DRIFT
ST	cD1
MU	Cos S
ST	Acc2
LD	cDv
MU	Sin S
AD	Acc2
ST	cD1'
LD	cD1
MU	Sin S
ST	Acc2
LD	cDv
MU	Cos S
SB	Acc2
AD	E_{sf}

(Continued in the next column)

ST	E_q
LD	Sin S
MU	Cos C
ST	Acc2
LD	Cos S
MU	Sin C
AD	Acc2
MU	Sin D
DV	Cos E_{sf}
EN	ARSIN
AD	S_p
ST	Acc2
LD	cD1'
DV	Cos E_{sf}
AD	Acc2
ST	T_g
OU	OUTPUT 1
LD	E_q
OU	OUTPUT 2
LD	(t_z+DT)
OU	OUTPUT 3
LD	R_{gz}
OU	OUTPUT 4
LD	$\dot{B}_{ld}$

Contd...

OU	OUTPUT 5
LD	$\check{S}_d$
OU	OUTPUT 6

Total Number of Instructions in main program = 276

SUBROUTINE FOR "SIN" &"COS"

Entry for COS	ST	Acc2	PB	ST	Acc2
	LD	CT16		LD	CT19
	SB	Acc2		ST	SG
Entry for SIN	TP	PA	PD	LD	CT16
	AD	CT17		SB	Acc2
PA	ST	Acc2		TP	PC
	LD	CT16		LD	CT18
	SB	Acc2		SB	Acc2
	TP	PC		ST	Acc2
	LD	Acc2	PC	LD	Acc2
	SB	CT18		ST	THETA
	TP	PB		MU	Acc2
	CS	...		CS	...
	ST	Acc2		ST	Acc2
	LD	CT10		DV	CT20
	ST	SG		AD	CT10
	TP	PD		MU	Acc2

(continued in the next column)

Contd....

```
DV      CT21
AD      CT10
MU      Acc2
DV      CT22
AD      CT10
MU      Acc2
DV      CT23
AD      CT10
MU      THETA
MU      SG
RM      ...
```

Total number of instructions in Subroutine SIN/COS = 45

SUBROUTINE SQRT.

Algorithm used:

$$Y_{i+1} = \frac{Y_i}{2} + \frac{C}{2Y_i}$$

where $Y = \text{SQRT}(C)$

```
        ST      Acc2
        ST      ARG
LS      LD      Acc2
        ST      Xo
        MU      CT8
        ST      Acc2
        LD      ARG
```

```
        MU      CT8
        ST      Acc2
        LD      ARG
        MU      CT8
        DV      X
                 o
        AD      Acc2
        ST      Acc2
        MU      CT24
        ST      EX
        LD      X
                 o
        SB      ACC2
        TP      PX
        CS      ...
PX      SB      EX
        TP      IS
        LD      Acc2
        RN      ...
```

Total No. of instructions = 21

<u>SUB ROUTINE ARSIN</u>

```
        TP      XA
        CS      ...
        ST      ARG
        LD      CT19
        ST      SG
```

```
        CS      ...
        TP      XB
XA      ST      ARG
        LD      CT10
        ST      SG
XB      LD      ARG
        MU      ARG
        ST      X0
LP      LD      EX
        MU      EX
        MU      X0
        ST      Acc2
        LD      EX
        AD      CT10
        ST      SG
        AD      CT10
        ST      EX
        MU      X1
        ST      X1
        LD      Acc2
        DV      X1
        ST      Acc2
        AD      ARG
        ST      ARG
        LD      Acc2
```

(continued in the next column)

```
        TP      XC
        CS      ...
XC      SB      CT25
        TP      LP
        LD      Acc2
        MU      SG
        RN      ...
```

Total No. of Instructions = 37

SYMBOL TABLE

No.	Symbol	No.	Symbol	No.	Symbol
1	DT	2	$\dot{R}$	3	$\dot{S}$
4	$\dot{B}_1$	5	R_p	6	S_p
7	T_s	8	S	9	DTD
10	E_{sp}	11	B_{wr}	12	WV
13	E_q'	14	$\ddot{R}$	15	dt
16	$\ddot{S}$	17	$\ddot{B}_1$	18	Sin S_p
19	Cos S_p	20	Sin S	21	Cos S
22	$\dot{B}_{1d}$	23	$\dot{S}_d$	24	w
25	$\dot{C}$	26	$\dot{B}_1\dot{C}$	27	$\dot{w}$
28	Sin C	29	Cos C	30	R_{pw}
31	S_f	32	Cos S_f	33	Sin S_f
34	WVC	35	R_f	36	R_{fo}
37	t_f	38	R_f Cos D	39	Sin D
40	D	41	T_g	42	Bwg
43	Sin Bwg	44	Cos Bwg	45	KW
46	TE	47	CT1	48	CT2
49	CT3	50	CT4	51	CT5
52	CT6	53	CT7	54	CT8
55	CT9	56	CT10	57	CT11
58	CT12	59	CT13	60	CT14
61	CT15	62	CT16	63	CT17
64	CT18	65	CT19	66	CT20

Contd.....

SYMBOL TABLE (Contd.)

No.	Symbol	No.	Symbol	No.	Symbol
67	CT21	68	CT22	69	CT23
70	CT24	71	CT25	72	DRIFT
73	t_z	74	t_{zo}	75	(t_z+DT)
76	$R_z Sin\ D_z$	77	R_{gz}	78	$Cos\ E_{sp}$
79	E_{sf}	80	RWC	81	cDv
82	cDl	83	cDl'	84	E_q
85	SG	86	THETA	87	ARG
88	X_o	89	X_1	90	EX
91	(TE+DIP)				

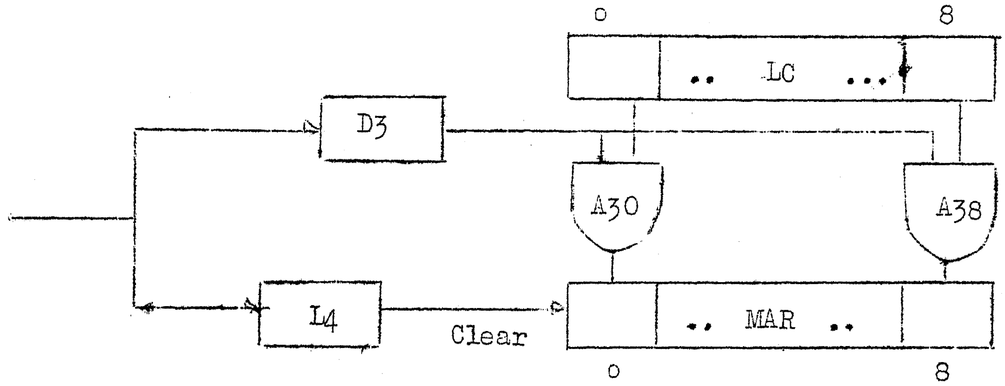
APPENDIX III

CONTROL CIRCUITS TO EXECUTE MICRO/MACRO-OPERATIONS

A gate level design of the control circuits to execute the micro and macro-operations are given below. The following notations are used to indicate various logical blocks.

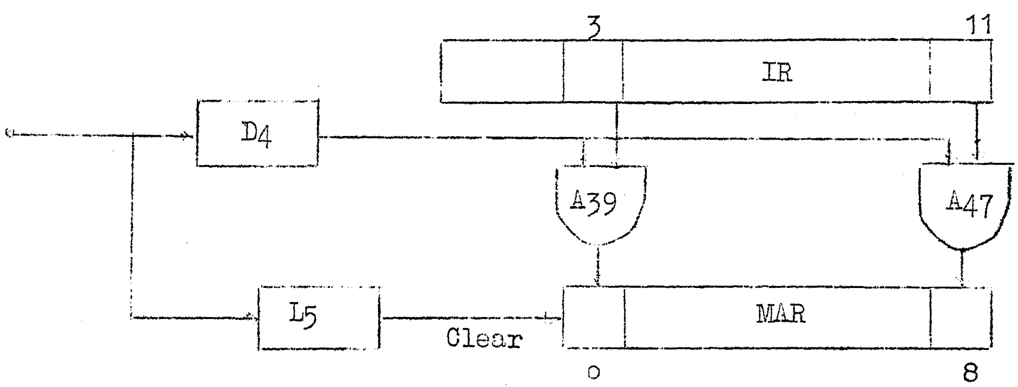
1) FF : Flip flop
2) L : Latch
3) D : Delay circuits
4) A : AND gates
5) R : OR gates
6) C : Counter
7) CLK: Clock pulse
8) MPX: Multiplexer
9) RS : Reset flip flop/clear the register
10) S : Set flip flop
11) AC: : Addition complete
12) MC : Multiplication complete
13) DC : Division Complete

M1:



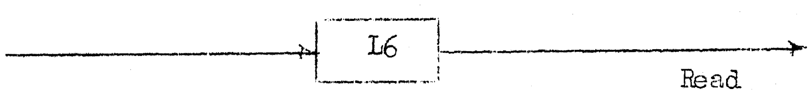
When the M1 line is pulsed, latch L4 clears MAR. The input pulse is delayed by D3 till the MAR is cleared. Output of D3 is given to AND gates A30 through A38 which transfer the contents of LC to MAR.

M2:

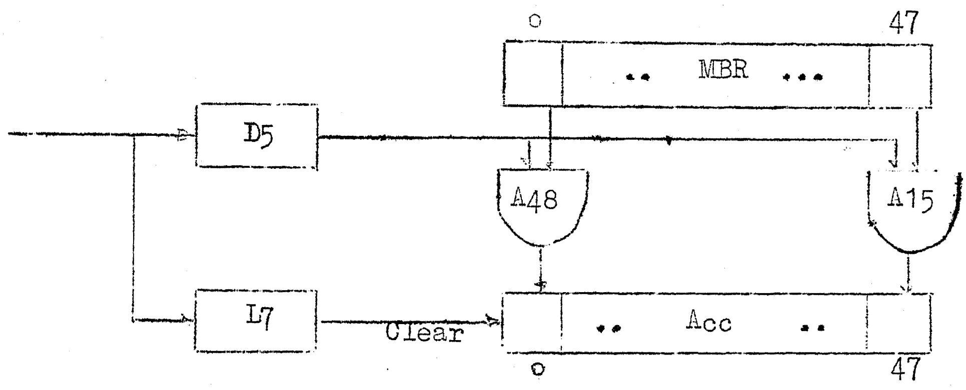


When the input line of M2 is pulsed, D4 delays the pulse till L5 clears MAR. Output of D4 enables transferring of the contents of the address part of IR to MAR by AND gates A39 through A47.

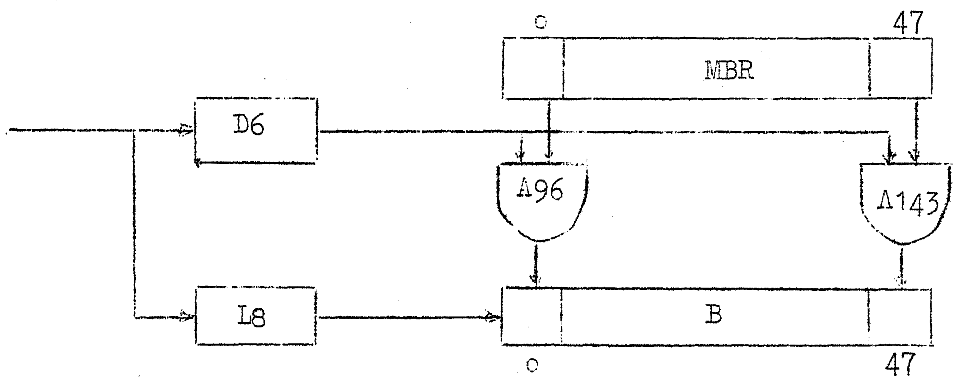
M3:



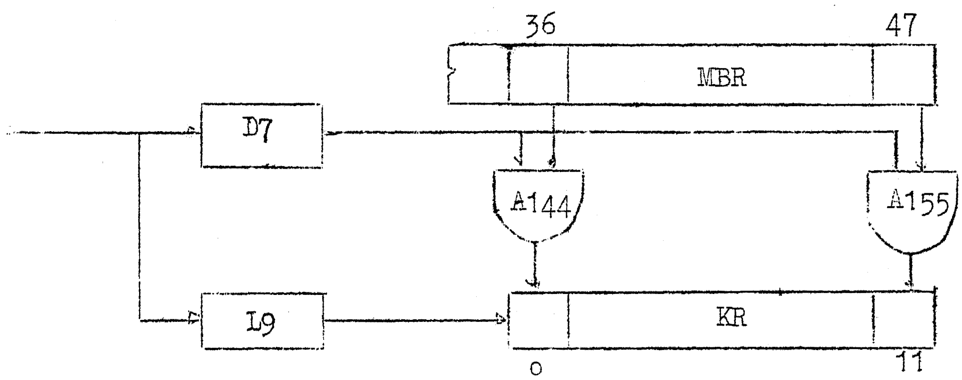
M4:



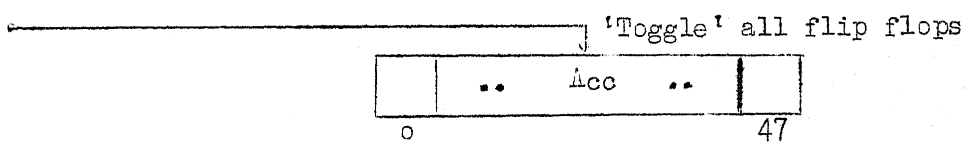
M5:



M6:

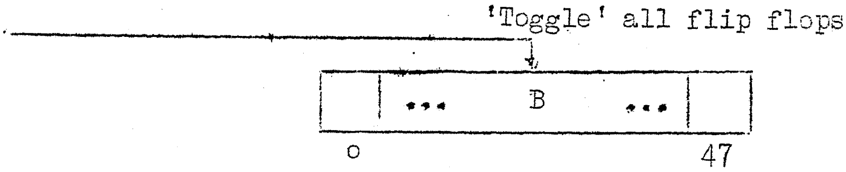


M7:

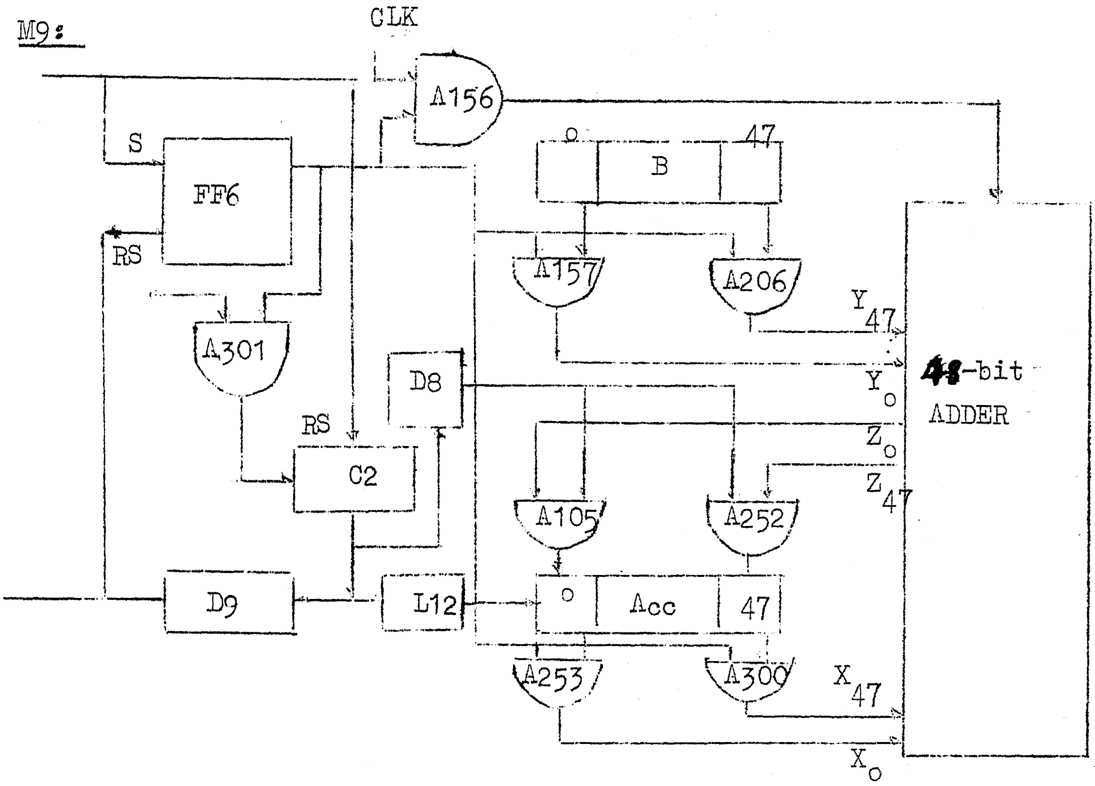


When the input line of M7 is pulsed, L10 pulses the J and K inputs of all flip flops of Acc which toggles all flip flops in Acc, effecting 1's complementation.

M8:

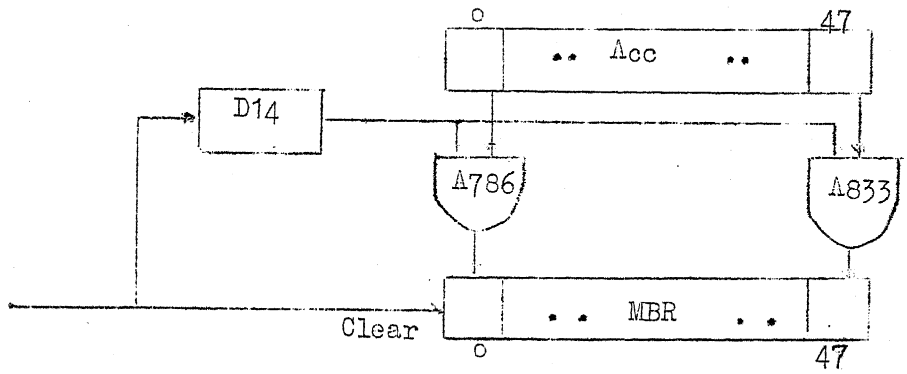


M9:

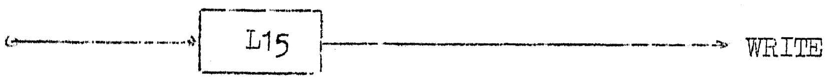


When the input line of M9 is pulsed, FF6 is set and the 'n' counter is reset. 1 output of FF6 connects Acc and B registers to the Adder through AND gates A158 through A301 and the Addition commences. The 'n' counter C2 starts counting the pulses. When the counter has counted upto 'n', output of C2 resets FF6 and gives addition complete (AC) signal to the MSG.

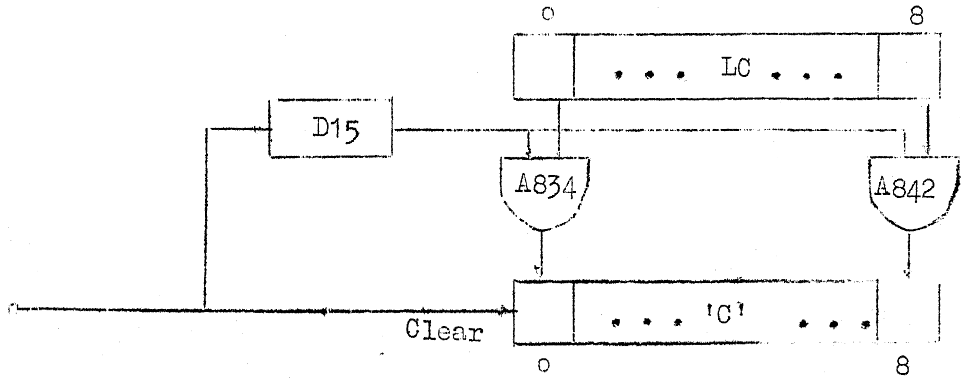
M12:



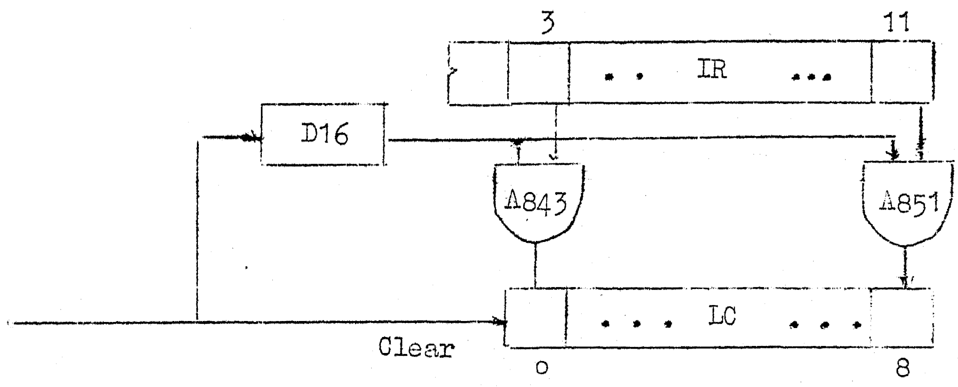
M13:



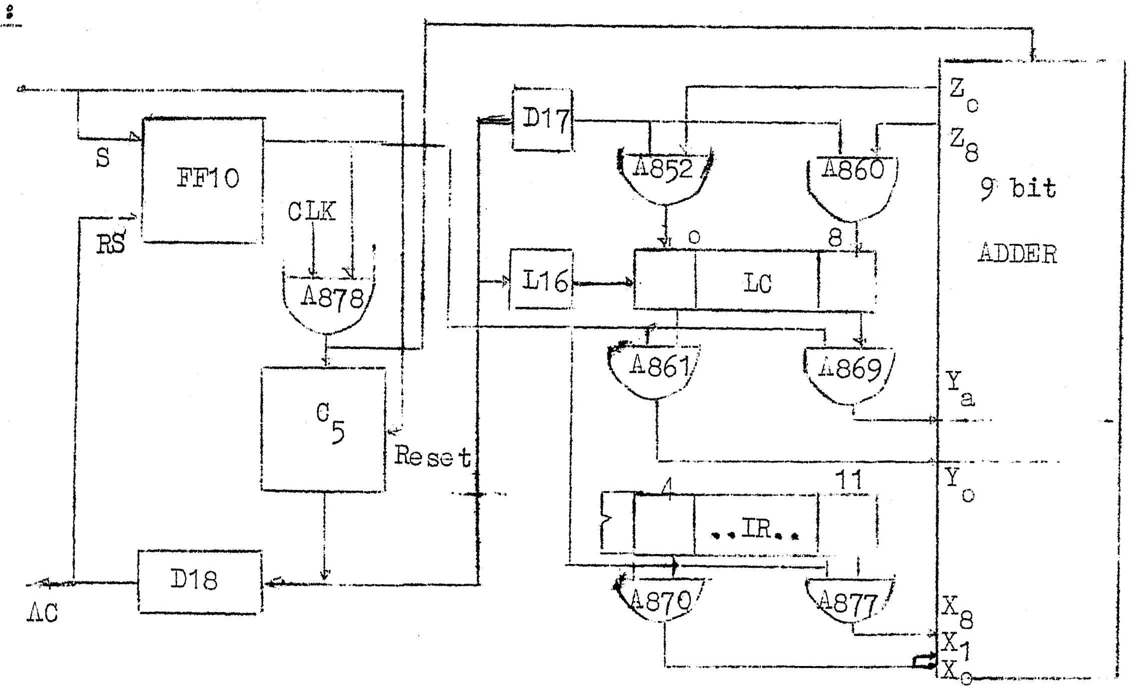
M14:



M15:

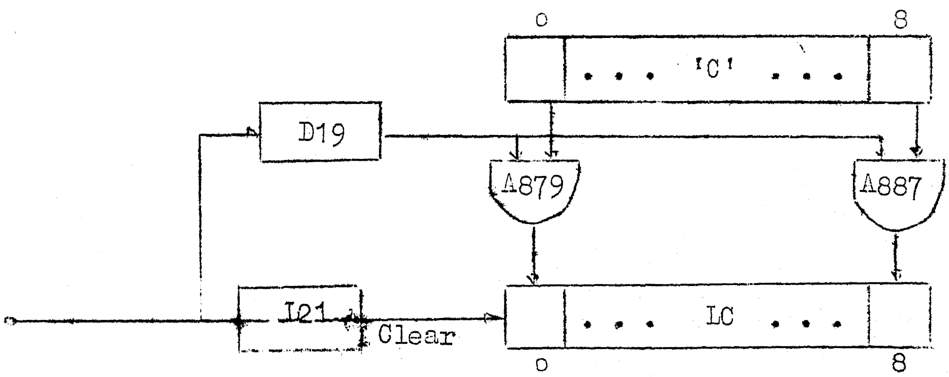


M16:

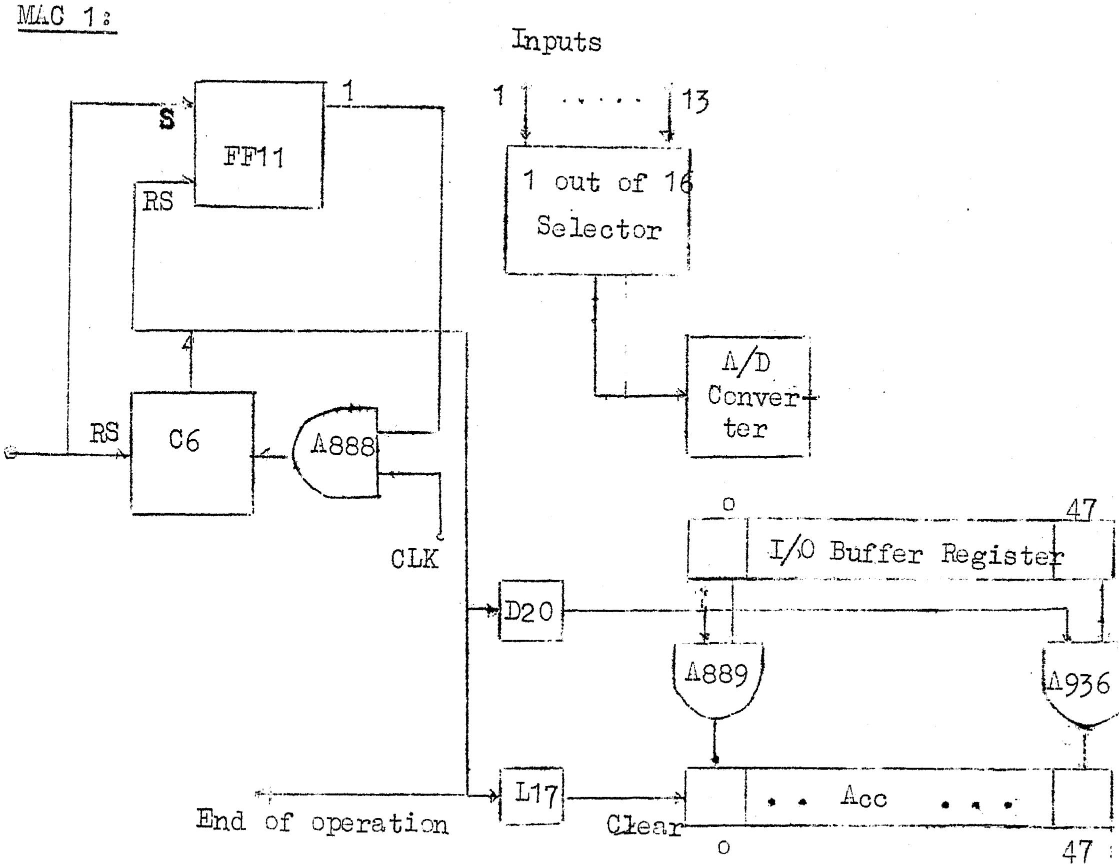


When input line of M16 is pulsed, FF10 is set and the 'n' counter is rest. 1 output of FF10 connects the LC and bits 4 through 11 of IR to the 9 bit parallel Adder, through the AND gates A861 through A877. The addition commences and C5 starts counting. the clock pulses. When the counter has counted upto 'n', output of C5 resets FF10 and gives addition complete signal to MSG8. Here LC has 9 bits where as the signed displacement i.e. bits 4 through 11 of IR is of 8 bits. Hence bit 4 of IR is connected to i.e. X_0 and X_1 the two most significant bits of the X input to the 9 bit Adder.

M17:

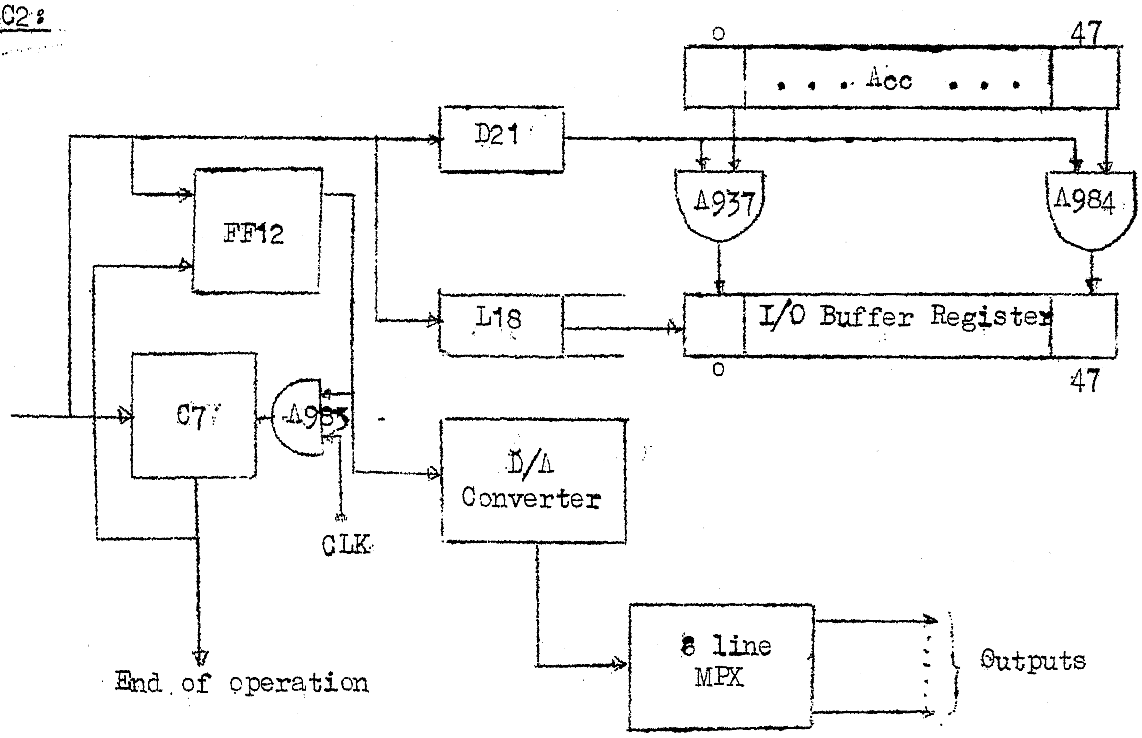


MAC 1:



When the input line of MAC1 is pulsed, FF11 is set and C6 is reset. 1 output of FF11 enables the 1 out of 16 selector to select the input line as indicated by bits 8 through 11 of IR and connects it to the A/D converter. The C6 starts counting. When C6 has counted upto 'n', where 'n' is the number of clock time A/D conversion takes, output of C6 resets FF11 and shifts the contents of I/O Buffer Register to Acc.

MAC2:



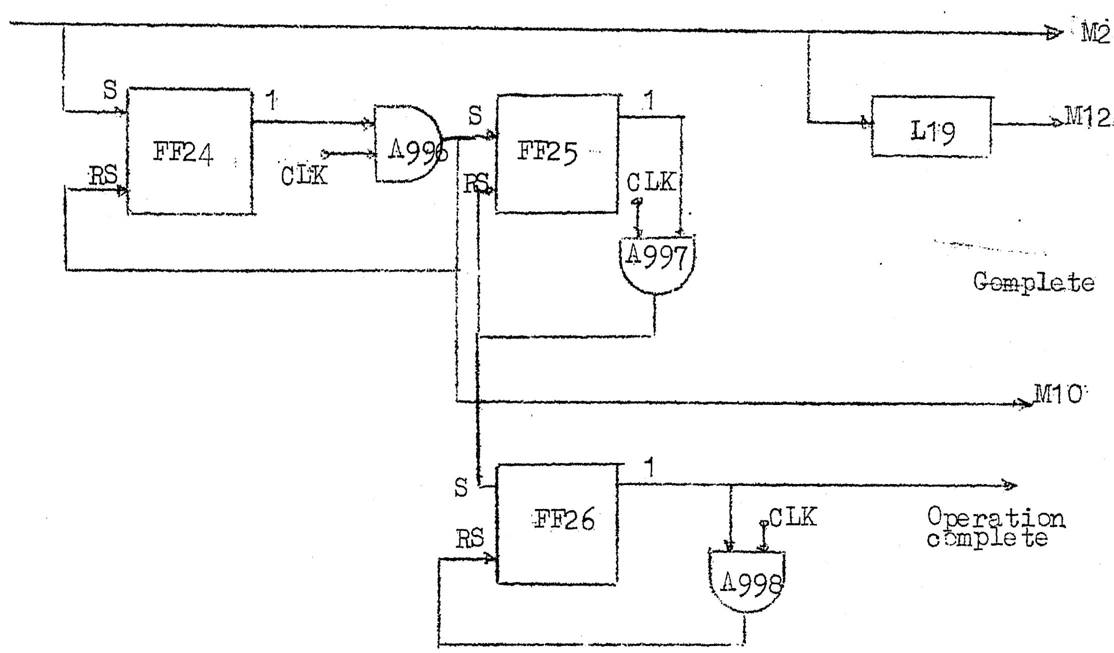
When input line of MAC2 is pulsed FF12 is set, C7 is reset and the contents of Acc are shifted to I/O Buffer register. 1 output of FF12 enables the D/A converter to convert the contents of I/O Buffer Register into Analog signal. The 8 line Multiplexer (MPX) connects the output of the D/A converter to the output line indicated by bits 9 through 11 of IR. When the D/A conversion is over, output of C7 resets FF12 and gives end of operation signal.

APPENDIX - IV

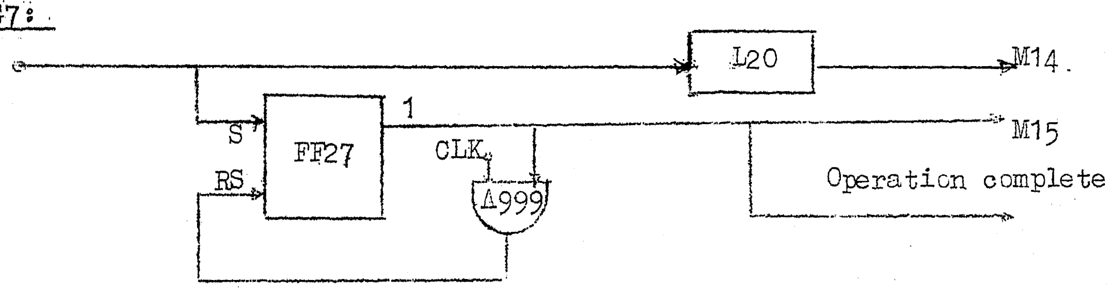
MICRO SEQUENCE GENERATORS

Conventional Circuits for Micro Sequence Generators (MSG) are given below. Notations given in Appendix - III are used.

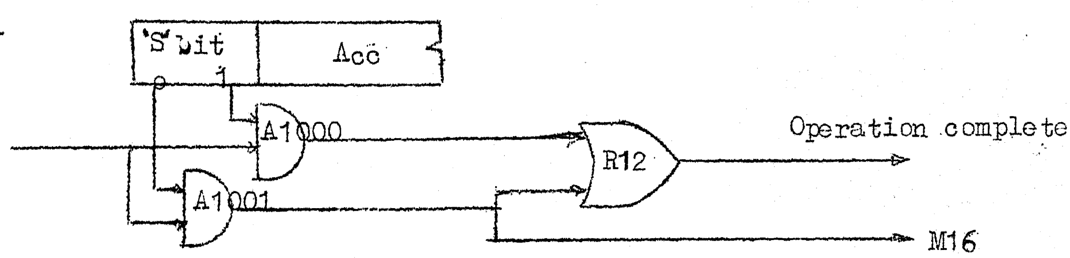
MSG6:



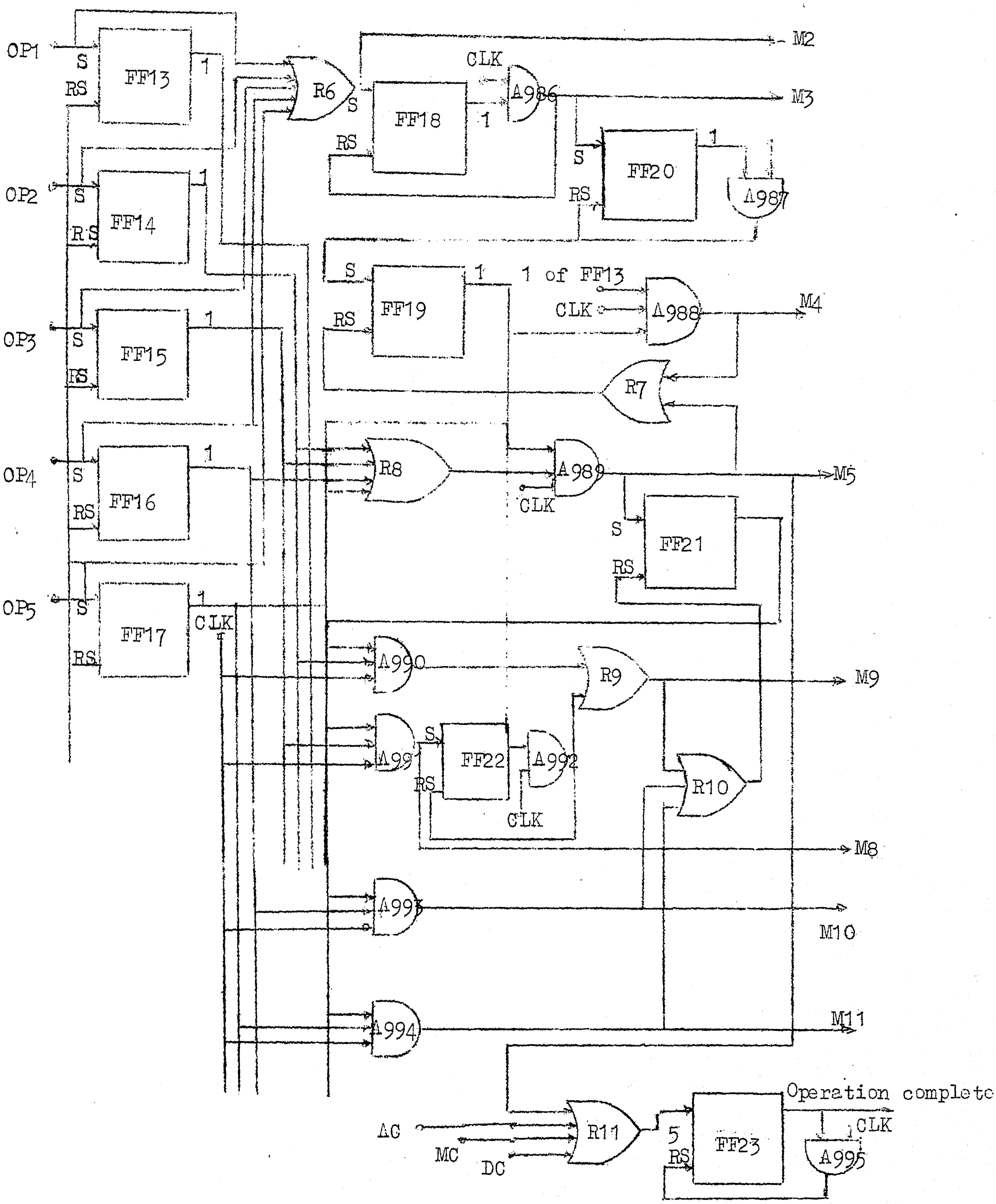
MSG7:



MSG8:



MSG1 through MSG5



LIST OF REFERENCES

(1) DANIEL, C & WOOD, F.S: "Fitting Equations to Data"
John Wiley & Sons Inc. 1971. [19, 20]

(2) KRAFT, A.ERICKE: "Space Flight, Vol.II Dynamics"
Principles of Guided Missile Design Series
Merril Editor, McGraw Hill Book Company, 1954. [12]

(3) MORRISE, R.L. & MILLER, J.R., "Designing with TTL Integrated Circuits", Texas Instruments Inc. 1966. [36,38]

(4) RILEY, W.B: "Electronic Computer Memory Technology"
McGraw Hill Book Company, 1971. [40]

(5) Fairchild Semiconductor Integrated Circuit Data Catalog, 1970. [46,58]

GENERAL REFERENCES

(1) LOCKE: "Guidance" Principles of Guided Missile Design Series Merril Editor. 1958. [19, 20]

(2) SHAPIRO : "Predicting Ballistic Missile Trajectories from Radar Measurements" Principles of Guided Missile Design Series Merril Editor. 1958. [12]

(3) SOBEL, H.S: "Introduction to Digital Computer Design". Addison Wesley Publishing Company 1970. [36, 38]

(4) WOLBERG, J.R: Prediction Analysis, Von Nortrand, Princeton, N.J. 1967. [40]

(5) Integrated Circuits Catalog for Design Engineers - Texas Instruments. Inc. [46, 58]

GLOSSARY

Acc: Accumulator Register

B_f: Future bearing of target

$\dot{B}_l$: Rate of change of bearing in lateral plane

B_{wr}: Relative Wind bearing

C_d: Combined drag coefficient

DB: Error in prediction of bearing

DE: Error in prediction of elevation

DR: Error in prediction of range

E_f: Predicted future elevation

E_g: Gun elevation

F_b: Improvement factor for bearing.

F_e: Improvement factor for elevation

F_r: Improvement factor for range

IC: Integrated circuit

IR: Instruction Register

LC: Location Counter

LSI: Large Scale Integration

MAR: Memory Address Register

MBR: Memory Buffer Register

MQ: Multiplier Quotient Register

MSG: Micro-Sequence Generators

MSI: Medium Scale Integration

MTBF: Mean time between failures

P_v: Projective velocity

R_f: Predicted future range

R_p: Present range

R: Rate of change of range

S: Rotation of sight line

S_p: Present elevation

$\dot{S}$: Rate of change of elevation

SSI: Small Scale Integration

T_g: Gun training

T_s: Present bearing

TTL: Transistor-transistor logic

WV: Wind velocity

X: Angle of approach

Y: Diving angle of aircraft

dt: Sampling to interval

g: Acceleration due to gravity

t_f: Predicted time of flight of the Shell

v: Velocity of target.

w: Angular rate of the target